

ICES REPORT 12-05

January 2012

An Isogeometric Design-through-analysis Methodology based on Adaptive Hierarchical Refinement of NURBS, Immersed Boundary Methods, and T-spline CAD Surfaces

by

D. Schillinger, L. Dede, M.A. Scott, J.A. Evans, M.J. Borden, E. Rank, T.J.R. Hughes



The Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Reference: D. Schillinger, L. Dede, M.A. Scott, J.A. Evans, M.J. Borden, E. Rank, T.J.R. Hughes, An Isogeometric Design-through-analysis Methodology based on Adaptive Hierarchical Refinement of NURBS, Immersed Boundary Methods, and T-spline CAD Surfaces, ICES REPORT 12-05, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, January 2012.

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 23 JAN 2012		2. REPORT TYPE		3. DATES COVERED 00-00-2012 to 00-00-2012	
4. TITLE AND SUBTITLE An Isogeometric Design-through-analysis Methodology based on Adaptive Hierarchical Refinement of NURBS, Immersed Boundary Methods, and T-spline CAD Surfaces				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Texas at Austin, Institute for Computational Engineering and Sciences, Austin, TX, 78712				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT We explore hierarchical refinement of NURBS as a basis for adaptive isogeometric and immersed boundary analysis. We use the principle of B-spline subdivision to derive a local refinement procedure which combines full analysis suitability of the basis with straightforward implementation in tree data structures and simple generalization to higher dimensions. We test hierarchical refinement of NURBS for some elementary fluid and structural analysis problems in two and three dimensions and attain good results in all cases. Using the B-spline version of the finite cell method, we illustrate the potential of immersed boundary methods as a seamless isogeometric design-through-analysis procedure for complex engineering parts defined by T-spline CAD surfaces, specifically a ship propeller and an automobile wheel. We show that hierarchical refinement considerably increases the flexibility of this approach by adaptively resolving local features.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 60	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

An Isogeometric Design-through-analysis Methodology based on Adaptive Hierarchical Refinement of NURBS, Immersed Boundary Methods, and T-spline CAD Surfaces

Dominik Schillinger^{a,b,*}, Luca Dedè^b, Michael A. Scott^b,
John A. Evans^b, Michael J. Borden^b, Ernst Rank^a, Thomas J. R. Hughes^b

^a*Lehrstuhl für Computation in Engineering, Technische Universität München, Germany*

^b*Institute for Computational Engineering and Sciences, The University of Texas at Austin, USA*

Abstract

We explore hierarchical refinement of NURBS as a basis for adaptive isogeometric and immersed boundary analysis. We use the principle of B-spline subdivision to derive a local refinement procedure, which combines full analysis suitability of the basis with straightforward implementation in tree data structures and simple generalization to higher dimensions. We test hierarchical refinement of NURBS for some elementary fluid and structural analysis problems in two and three dimensions and attain good results in all cases. Using the B-spline version of the finite cell method, we illustrate the potential of immersed boundary methods as a seamless isogeometric design-through-analysis procedure for complex engineering parts defined by T-spline CAD surfaces, specifically a ship propeller and an automobile wheel. We show that hierarchical refinement considerably increases the flexibility of this approach by adaptively resolving local features.

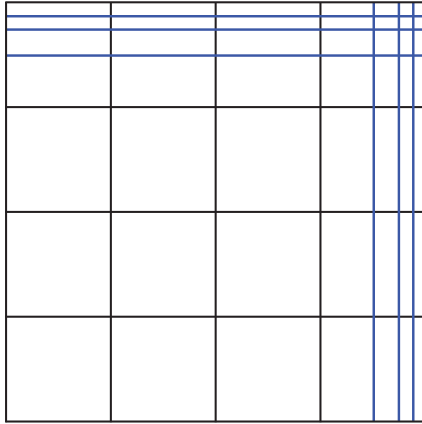
Keywords: Isogeometric analysis, hierarchical refinement, adaptivity, NURBS, immersed boundary analysis, finite cell method, T-spline surfaces

1. Introduction

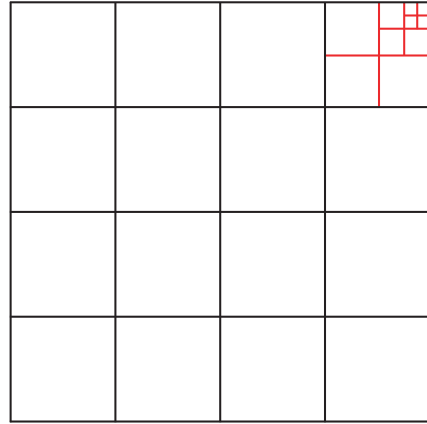
Isogeometric analysis (IGA) was introduced by Hughes and co-workers [1] to bridge the gap between computer aided design (CAD) and finite element analysis (FEA). Its core idea is to use the same basis functions for the representation of geometry in CAD and the approximation of solutions fields in FEA. This strategy bypasses the mesh generation process required for standard FEA and supports a tightly connected interaction between CAD and FEA tools [2–4], which could potentially reduce the time required for the analysis of complex engineering designs by up to 80% [1, 5]. In addition, it has been shown that the use of a smooth, higher-order geometric basis is superior to

*Corresponding author;

Lehrstuhl für Computation in Engineering, Technische Universität München, Arcisstr. 21, 80333 München, Germany;
Phone: +49 89 289 25116; Fax: +49 89 289 25051; E-mail: schillinger@bv.tum.de



(a) Standard h -refinement by knot insertion leads to global refinement due to the tensor product structure of NURBS.



(b) What is required is truly local refinement, concentrated around the area at issue without “spreading out” globally.

Figure 1: Example of a mesh of 4×4 NURBS elements that should be refined on the upper right corner.

standard C^0 discretizations [6]. This has been demonstrated for a variety of application areas such as structural vibrations [5, 7], incompressibility [8–10], shells [11–13], fluid-structure interaction [14, 15], turbulence [16–18], phase fields [19, 20], contact [21, 22], fracture [23] and optimization [24, 25]. From a practical perspective, the use of finite element forms based on Bézier extraction allow for a simple integration of smooth, higher-order bases (i.e. B-splines, NURBS, and T-splines) into existing finite element codes [26, 27].

1.1. Local refinement in isogeometric analysis

Local mesh refinement is commonly used in FEA to resolve local features. For example, an adaptive local refinement algorithm can be used to resolve internal and boundary layers in advection dominated flows and stress concentrations in structures. Due to their relative simplicity and ubiquity in today’s CAD tools, isogeometric analysis has been largely based on non-uniform rational B-splines (NURBS). However, in contrast to the standard nodal basis, a multivariate NURBS basis does not provide a natural possibility for local mesh refinement [5, 28–30]. Figure 1 illustrates this issue for the example of a simple bivariate NURBS patch. Due to its rigid tensor product structure, refinement of NURBS is a global process that propagates throughout the domain.

The concept of hierarchical refinement of B-splines was introduced by Forsey and Bartels for local surface refinement in CAD [31, 32] and later adopted by Höllig and co-workers for local mesh refinement in B-spline finite elements [33–35]. In the framework of isogeometric analysis, hierarchical refinement of NURBS has recently attracted increasing attention [36, 37] due to the following important advantages. First, hierarchical B-splines rely on the principle of B-spline subdivision [38, 39], which makes it possible to maintain linear independence throughout the refinement process. In addition, the maximal smoothness of NURBS is maintained in a hierarchically refined

basis. Second, since hierarchical B-splines rely on a local tensor product structure, they can be easily generalized to arbitrary dimensions. The rigidity and simplicity of the tensor product structure also facilitates automation of the refinement process. Third, a hierarchical organization of a basis can be directly transferred into a tree-like data structure, which is a well-established concept in computer science [40–42] and allows for a straightforward implementation with manageable coding effort. Fourth, very similar refinement techniques based on a hierarchical split of standard finite element bases have existed in the FEA community for a long time (see for example [43–46]), which can help one to become familiar with hierarchical B-spline refinement. In the first part of this paper, we explore the behavior of hierarchical B-splines and NURBS in the context of IGA. Specifically, we develop the ideas from both a theoretical and algorithmic viewpoint with an emphasis on demanding applications in both solids and fluids.

We note that there are alternative approaches for local refinement in isogeometric analysis. An analysis-suitable class of T-splines were recently introduced which are linearly independent, form a partition of unity, and can be refined in a highly localized manner [30, 47, 48]. An important distinction between local refinement of T-splines and the hierarchical methods presented in this paper is that T-spline local refinement is performed on a single hierarchical “level” and all control points have a similar influence on the shape of the surface. This is critical for design, but of less importance for analysis, where a hierarchy of refinements can be used to effectively resolve local features in a finite element solution. PHT-splines [49–52] are a further geometry representation that naturally accommodates local mesh refinement. However, PHT-splines do not obtain the maximal smoothness of B-splines, NURBS and T-splines that has been shown to be beneficial in both design and analysis.

1.2. T-splines as an isogeometric design-through-analysis enabling technology

At this point in time, based on recent advances and understanding of T-spline technology [28–30, 47, 48, 53, 54], it is our opinion that bi-cubic T-spline surface modeling has reached sufficient maturity that it may be considered a nearly complete isogeometric engineering design-through-analysis enabling technology. Currently, watertight parameterizations of surfaces can be constructed for geometrically and topologically complex engineering designs that can be used directly as finite element meshes in shell structural analysis and boundary element analysis of three-dimensional solids, avoiding the necessity of geometry repair (e.g., eliminating gaps and overlaps of NURBS patches) and feature removal of CAD files, from which meshes are usually generated. At the same time we do not wish to imply that research on T-spline surfaces is finished, quite the contrary. There are still numerous opportunities for improvements and deepening understanding, such as, for example, generalization to arbitrary polynomial degree and mixed-degree T-splines, efficient quadrature rules, more efficient local refinement schemes, development of convergence proofs in integral norms, and perhaps most importantly, achieving optimal convergence rates when extraordinary points are present. Nevertheless, analysis-suitable T-spline surfaces exist for many applications and may be

considered an initial instantiation of the vision of isogeometric analysis.

However, there are many engineering designs that require full, three-dimensional (i.e., trivariate) parameterizations. Presently, only partial isogeometric solutions to this problem exist (see, e.g., Zhang and co-workers [54–56]). This being the case, it is worthwhile to consider if the existing T-spline surface modeling capability can somehow be utilized to facilitate the analysis of three-dimensional solid and fluid domains. An obvious opportunity exists in the framework of so-called immersed boundary and interface methods, also known as fictitious domain or embedded domain methods. We will not attempt to review the immersed boundary methods literature here. Suffice it to say it is large and growing. A recent textbook by Li and Ito [57] is quite comprehensive and contains many references. There has also been a recent surge of interest in immersed methods, due to their ability to address particularly complex moving boundary and interface problems. The traditional philosophy of immersed boundary methods is to employ simple, mesh-aligned numerical schemes, such as finite differences on a uniform Cartesian grid, and to treat the cells that are cut by boundaries and/or interfaces with some special, and often ad hoc, technique. This view has been enhanced in recent years to include adaptive, locally refined meshes, but it is generally acknowledged that the fundamental problem facing immersed boundary methods is to stably and accurately treat the cut cells. Clearly, immersed methods naturally give rise to a “staircase” representation of boundaries and interfaces that needs to be improved in an appropriate way to achieve acceptably accurate results. A common criticism of immersed methods is the inability to accurately represent computed boundary and interface quantities (e.g., stress and heat flux), and sharp layers at boundaries and interfaces.

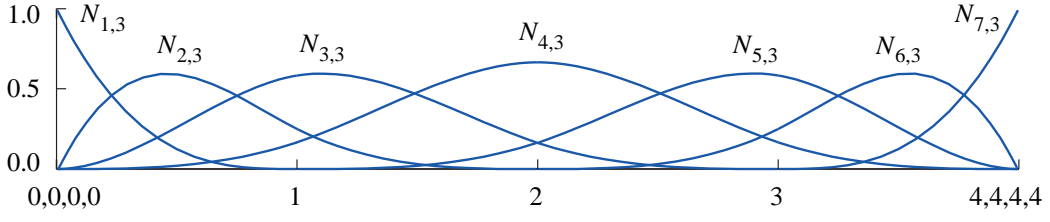
1.3. The finite cell method

Recently, a new strategy for treating boundaries and interfaces has been developed within the finite cell method (FCM) [58–60]. The rectilinear volumetric mesh may, or may not, be locally refined in the vicinity of boundaries or interfaces. Obviously, local refinement helps, but this is not the essential issue. The unique feature of the finite cell method is to create a highly refined quadrature mesh of sub-cells surrounding the boundary and interfaces. It needs to be emphasized that the sub-cells are utilized for numerical integration only and the basis functions are not defined with respect to them, but rather to the original coarser mesh. The resolution of boundary and interface phenomena is the responsibility of the refined quadrature mesh. Quadrature points within the sub-cells are tagged as being either inside the physical domain, or outside it, that is, being in the fictitious part of the domain, and treated accordingly. The procedure is very simple, but has exhibited some remarkable properties. These are: optimal convergence rates in integral norms have been demonstrated for uniformly h -refined finite element meshes, exponential convergence has been attained for p -refinement of uniform finite element meshes, and point-wise convergence of derivative quantities has been achieved up to, and on, the boundaries within the cut cells. In our opinion, this represents a major step forward as we know of no other immersed boundary method

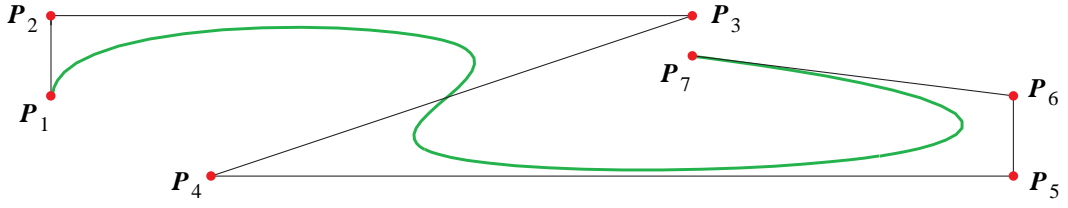
that has achieved these attributes. Thus, the finite cell method, when combined with the geometric modeling capability of T-spline surfaces, provides a pathway to the isogeometric design-through-analysis of many problems involving complex, real-world engineering objects. These technologies exist today and it is a primary aim of this paper to demonstrate their efficacy in the solution of three-dimensional structural analysis problems of real engineering designs, specifically, a ship propeller and an automobile wheel. Both these structures are impossible to classify within the traditional categories of structural analysis and solid mechanics models, that is, beams, plates, shells, and solids. Both have very thin shell-like regions, and also very thick solid-like regions, with transition regions in between. Thus, the only analysis possibility is to utilize full three-dimensional solid representation but, due to the complexity of these structures, it is exceedingly difficult to generate quality unstructured meshes with even the most advanced meshing tools available. On the other hand, the combination of T-spline surface models, the finite cell method, and hierarchically refined NURBS provides a complete design-through-analysis methodology that can be fully automated. We further wish to note that “trimming”, which is ubiquitous in engineering CAD, can also be easily accommodated within the finite cell method. A trimmed NURBS or T-spline surface design can be directly utilized as an isogeometric shell mesh in conjunction with the finite cell method, as has been demonstrated in [61, 62].

1.4. Structure and content of the paper

After a brief review of B-spline and NURBS basis functions in Section 2, we present in detail a hierarchical refinement scheme for axis-aligned B-splines for the representation of cuboidal geometries, drawing on ideas of hierarchical B-splines, the *hp-d* adaptive approach and established hierarchical refinement of standard nodal based FEA. Section 3 focuses on concepts and a basic illustration for a simple 1D example, and Section 4 illustrates hierarchical refinement in multiple dimensions and outlines the main aspects of an efficient implementation. Section 5 generalizes the hierarchical refinement principle to NURBS for the representation of arbitrary geometries. Section 6 shows the validity and efficiency of hierarchical refinement of NURBS for a range of two- and three-dimensional benchmark examples from fluid and structural mechanics. In Section 7, a short summary of the recently introduced B-spline version of the finite cell method is given, which combines the higher-order continuity of a B-spline approximation with the simple geometry handling of immersed boundary methods. In Section 8, we illustrate its potential as a seamless IGA design-through-analysis procedure for complex engineering parts and assemblies. We present two applications, a ship propeller and a rim of an automobile wheel, where we demonstrate that hierarchical refinement considerably increases the flexibility of immersed boundary methods by adaptively resolving local features in geometry and solution fields. In Section 9, we close our discussion with a brief summary, conclusions and suggestions for future research.



(a) Cubic B-spline patch with interpolatory ends.



(b) B-spline curve generated from the above basis using control points P_i .

Figure 2: Example of cubic B-spline basis functions and a corresponding B-spline curve in 2D.

2. B-spline and NURBS basis functions

We start with a brief review of some technical aspects of B-spline and NURBS bases for isogeometric analysis, placing particular emphasis on the features important for understanding the concepts and algorithms of their hierarchical refinement. Readers interested in a broader and more detailed introduction are referred to the fundamental works of Hughes and co-workers [1, 5] for isogeometric analysis and to Piegl and Tiller [63], Rogers [64] and Farin [65] for a comprehensive review of the underlying geometric concepts and algorithms.

2.1. Univariate B-splines

A B-spline basis of degree p is formed from a sequence of knots called a knot vector $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$, where $\xi_1 \leq \xi_2 \leq \dots \leq \xi_{n+p+1}$ and $\xi \in \mathbb{R}$ is called a knot. A univariate B-spline basis function $N_{i,p}(\xi)$ is defined using a recurrence relation, starting with the piecewise constant ($p = 0$) basis function

$$N_{i,0}(\xi) = \begin{cases} 1, & \text{if } \xi_i \leq \xi \leq \xi_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

For $p > 0$, the basis function is defined using the Cox-de Boor recursion formula

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad (2)$$

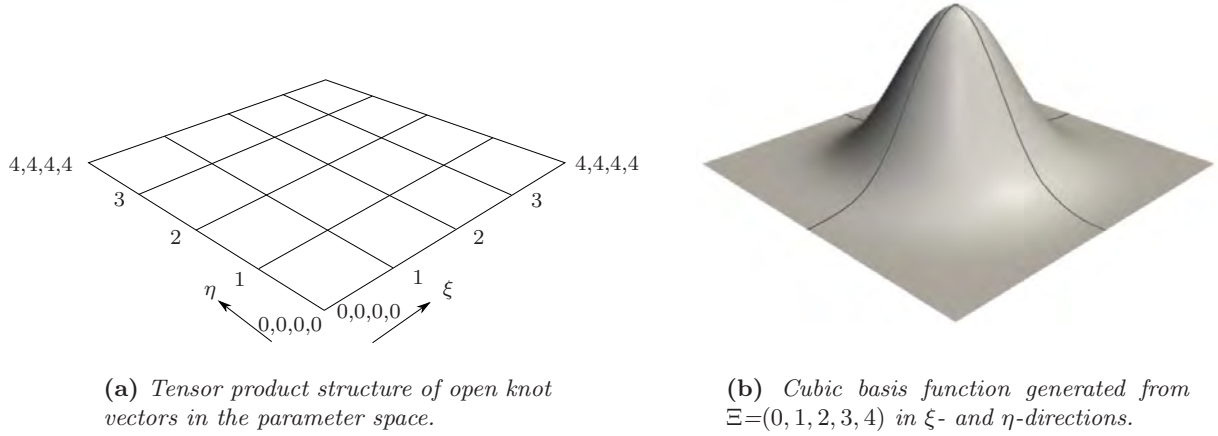


Figure 3: Bivariate cubic knot spans and a corresponding uniform B-spline basis function.

A repeated knot in Ξ is said to have multiplicity k . In this case, the smoothness of the B-spline basis is C^{p-k} at that location. Figure 2a illustrates a B-spline basis of polynomial degree $p = 3$ and knot vector $\Xi = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$, where knots at the beginning and the end are repeated to make the basis interpolatory.

Having constructed the corresponding basis functions, we can build a B-spline curve in d_s dimensions by a linear combination of basis functions

$$C(\xi) = \sum_{i=1}^n \mathbf{P}_i N_{i,p}(\xi) \quad (3)$$

where coefficients $\mathbf{P}_i \in \mathbb{R}^{d_s}$ are called control points. Piecewise linear interpolation of the control points defines the control polygon. An example generated from the B-spline basis shown in Figure 2a is provided in Figure 2b.

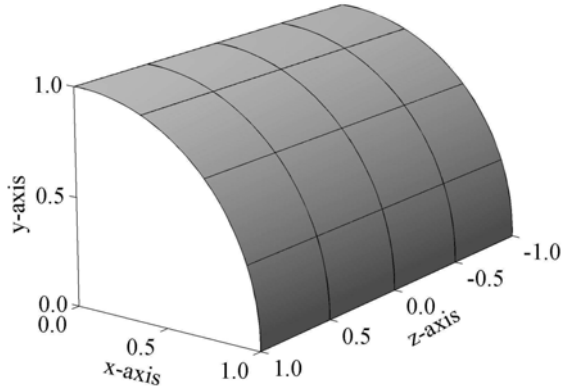
2.2. Multivariate B-splines

Multivariate B-splines are a tensor product generalization of univariate B-splines. We use d_s and d_p to denote the dimension of the physical and parameter spaces, respectively. Multivariate B-spline basis functions are generated from d_p univariate knot vectors

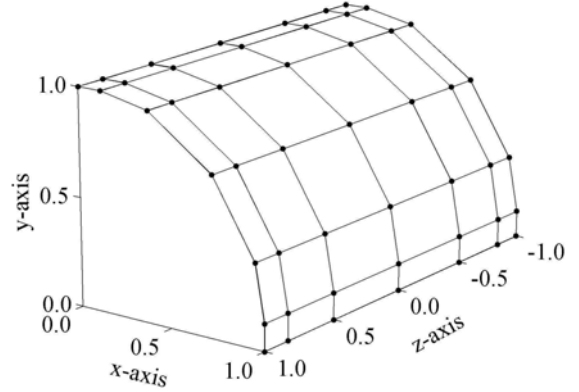
$$\Xi^\ell = \{\xi_1^\ell, \xi_2^\ell, \dots, \xi_{n_\ell+p_\ell+1}^\ell\} \quad (4)$$

where $\ell = 1, \dots, d_p$, p_ℓ indicates the polynomial degree along parametric direction ℓ , and n_ℓ is the associated number of basis functions. The resulting univariate B-spline basis functions in each direction ℓ can then be denoted by N_{i_ℓ, p_ℓ}^ℓ , from which multivariate basis functions $B_{i,p}(\boldsymbol{\xi})$ can be constructed as

$$B_{i,p}(\boldsymbol{\xi}) = \prod_{\ell=1}^d N_{i_\ell, p_\ell}^\ell(\xi^\ell) \quad (5)$$



(a) NURBS elements.



(b) Corresponding control mesh.

Figure 4: NURBS surface of a quarter of a cylinder, generated with the bivariate B-spline basis of Figure 3 and a set of suitable control points \mathbf{P}_i and weights w_i .

Multi-index $\mathbf{i} = \{i_1, \dots, i_{d_p}\}$ denotes the position in the tensor product structure, $\mathbf{p} = \{p_1, \dots, p_{d_p}\}$ indicates the polynomial degree, and $\boldsymbol{\xi} = \{\xi^1, \dots, \xi^{d_p}\}$ are the parametric coordinates in each parametric direction ℓ . A bivariate parametric space and B-spline basis function are shown in Figures 3a and 3b, respectively.

B-spline surfaces ($d_p = 2$) and solids ($d_p = 3$) are a linear combination of multivariate B-spline basis functions and control points in the form

$$\mathbf{S}(\boldsymbol{\xi}) = \sum_{\mathbf{i}} \mathbf{P}_{\mathbf{i}} B_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) \quad (6)$$

where the sum is taken over all combinations of multi-index \mathbf{i} . In the multivariate case, the control points $\mathbf{P}_{\mathbf{i}} \in \mathbb{R}^{d_s}$ form the so-called control mesh.

2.3. Non-uniform rational B-splines

NURBS can be obtained through a projective transformation of a corresponding B-spline in \mathbb{R}^{d_s+1} . Univariate NURBS basis functions $R_{i,p}(\xi)$ are given by

$$R_{i,p}(\xi) = \frac{w_i N_{i,p}(\xi)}{\sum_{j=1}^n w_j N_{j,p}(\xi)} \quad (7)$$

where $N_{i,p}(\xi)$ are polynomial B-spline basis functions and w_i are weights. Multivariate NURBS basis functions are formed as

$$R_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) = \frac{w_{\mathbf{i}} B_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi})}{\sum_{\mathbf{j}} w_{\mathbf{j}} B_{\mathbf{j},\mathbf{p}}(\boldsymbol{\xi})} \quad (8)$$

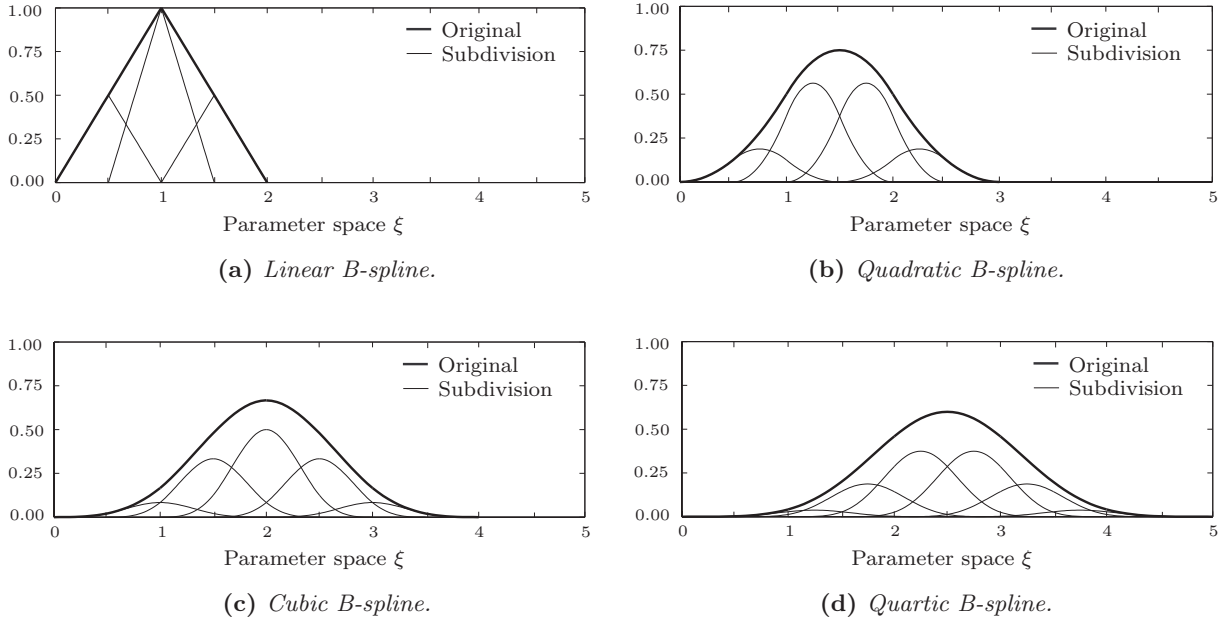


Figure 5: Subdivision of an original uniform B-spline into $p+2$ contracted B-splines of half the knot span width, illustrated for polynomial degrees $p=1$ through 4.

NURBS curves, surfaces and solids are then defined as

$$S(\xi) = \sum_i P_i R_{i,p}(\xi) \quad (9)$$

The NURBS example of Figure 4a represents a quarter of a cylinder exactly. It is generated by inserting the bivariate cubic B-spline basis of Figure 3 together with a suitable set of control points and weights into the NURBS Equations (8) and (9). Figure 4b shows the corresponding control mesh composed of the control points P_i . Suitable control points and weights can be derived in this simple case from [63–65] or for more complex examples from CAD tools such as Rhino [4, 66].

3. A concept for hierarchical refinement based on B-spline subdivision

In the following, we briefly review B-spline subdivision and show how this concept can be deployed to set up a hierarchical scheme for local refinement of B-spline basis functions, which combines an intuitive principle, full analysis suitability and straightforward implementation. We illustrate basic ideas of our hierarchical refinement strategy with a simple advection-diffusion example in one dimension.

3.1. Refinability of B-spline basis functions by subdivision

A remarkable property of uniform B-splines is their natural refinement by subdivision. For a univariate B-spline basis function N_p of polynomial degree p , the subdivision property leads to the

following two-scale relation [38, 39, 67]

$$N_p(\xi) = 2^{-p} \sum_{j=0}^{p+1} \binom{p+1}{j} N_p(2\xi - j) \quad (10)$$

where the binomial coefficient is defined as

$$\binom{p+1}{j} = \frac{(p+1)!}{j!(p+1-j)!} \quad (11)$$

In other words, a B-spline can be expressed as a linear combination of contracted, translated and scaled copies of itself, which is illustrated for B-splines of different polynomial degrees in Figure 5. Equation (10) does not hold for non-uniform B-splines with repeated knots, but similar subdivision rules can be constructed [68, 69].

Due to their tensor product structure, the generalization of subdivision to multivariate B-splines is a straightforward extension of Equation (10) and can be written as

$$B_{\mathbf{p}}(\boldsymbol{\xi}) = \sum_{\mathbf{j}} \left(\prod_{\ell=1}^d 2^{-p_\ell} \binom{p_\ell+1}{j_\ell} N_{p_\ell}(2\xi^\ell - j_\ell) \right) \quad (12)$$

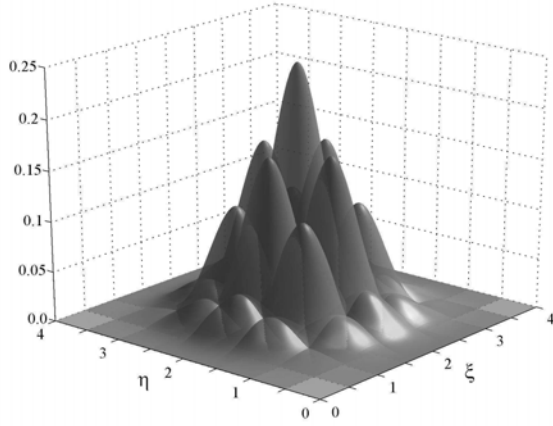
Following Section 2.2, multi-indices $\mathbf{j}=\{j_1, \dots, j_{d_p}\}$, $\mathbf{p}=\{p_1, \dots, p_{d_p}\}$ and $\boldsymbol{\xi}=\{\xi^1, \dots, \xi^{d_p}\}$ denote the position in the tensor product structure, the polynomial degree and the independent variables in each direction ℓ of the d_p -dimensional parameter space. Figure 6 illustrates the new basis functions resulting from the multivariate two-scale relation Equation (12) applied to the bivariate cubic B-spline of Figure 3b. The most widely known application of Equations (10) and (12) is the development of highly efficient subdivision algorithms for the fast and accurate approximation of smooth surfaces by control meshes in computer graphics [38, 39, 68, 69].

3.2. Construction of adaptive hierarchical approximation spaces

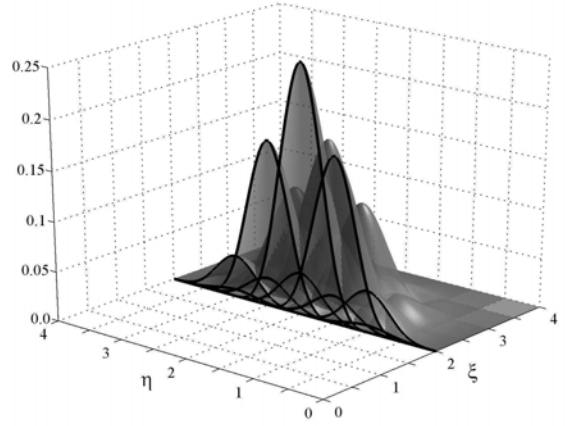
In the following, we will derive step by step a hierarchical scheme for local refinement of B-splines in one dimension, which combines concepts from B-spline subdivision, the *hp-d* adaptive approach [36, 70, 71] and existing hierarchical refinement techniques for B-spline finite elements [31, 33, 35] and for standard nodal based FEA [43, 45]. Our main goal is to arrive at a local refinement strategy, which maintains theoretical consistency, can be straightforward generalized to two and three dimensions and NURBS, but can be implemented in arbitrary dimensions with manageable coding efforts.

3.2.1. Element and basis viewpoints on refinement

Traditionally, refinement in FEA adopts an element viewpoint, since the centerpiece of standard mesh refinement consists of the geometric division of elements [44, 45]. Furthermore, an element



(a) Contracted, translated and scaled B-splines.



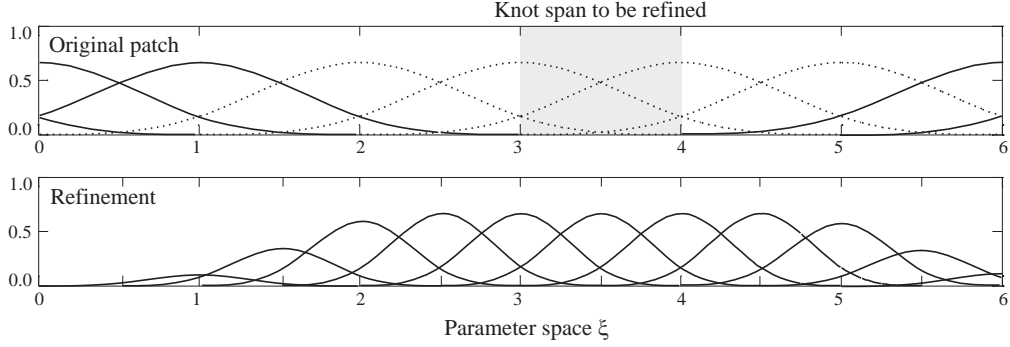
(b) Cut along $\xi=2.0$.

Figure 6: Subdivision of the bivariate cubic B-spline shown in Figure 3.

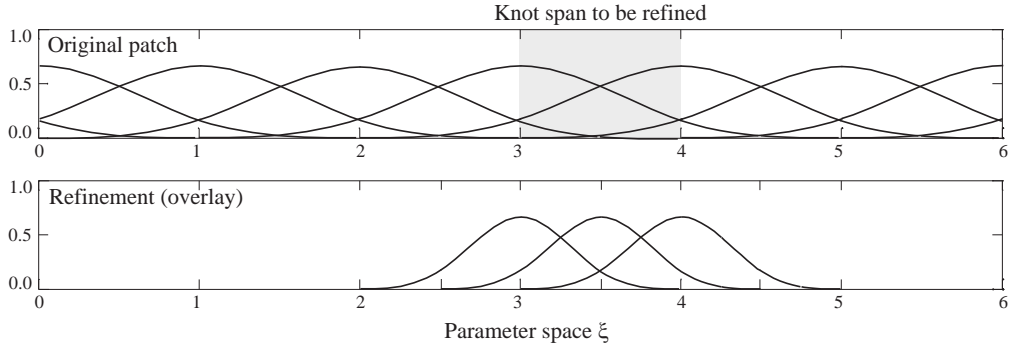
by element view accommodates most error estimators [72], which specify the error element wise, and naturally corresponds to the way finite element procedures are traditionally implemented. In isogeometric analysis, however, it is often more suitable to look at the complete basis, since an element centered viewpoint may obstruct an intuitive understanding of refinement principles for basis functions, which extend over several knot span elements [5, 44]. Therefore, we will adopt the element viewpoint, wherever we feel that aspects can be reasonably explained from there. Wherever it becomes too restrictive or interferes with the consistency of the hierarchical methods at issue, we switch to the more comprehensive basis viewpoint.

3.2.2. Two-level hierarchical refinement for one element

In a first step, let us define a nucleus operation, from which we start our development: the refinement of one knot span element. Figure 7 exhibits a portion of a B-spline patch, where the element in the center should be refined. A straightforward approach, introduced for B-spline FEA by Kraft [33] and Höllig [35] and recently extended to isogeometric analysis by Vuong and coworkers [37], is the application of the two-scale relation Equation (10) to all basis functions with support in the knot span element under consideration. Contracted B-splines resulting from the subdivision of different B-spline basis functions, but with the same support, are superposed by adding their scaling factors. This leads to full hierarchical B-spline basis functions in the center of the refinement, while hierarchical B-splines at the boundary are gradually decreased as shown in Figure 7a. However, the direct subdivision based refinement strategy results in a large spread of the refinement beyond the bounds of the knot span element under consideration, which obstructs the localization of refinement. Due to the increase of the spread with p , this is especially true for



(a) Subdivision refinement, splitting original B-splines with support in the element (dotted lines) according to the two-scale relation. The refined basis consists of all functions shown with solid lines.



(b) Hierarchical refinement, inspired by the hp -d adaptive approach [36]. The combination of the original patch and three contracted B-splines yields the refined basis.

Figure 7: Nucleus operation: the refinement of one knot span element, illustrated with cubic B-splines.

higher polynomial degrees. In addition, it requires a correct book-keeping and addition of scaling factors, which considerably complicates its implementation.

Therefore, we follow a different approach here and borrow the main idea of the hp -d adaptive approach, which was originally introduced for the p -version of the FEM [70] and successfully applied to B-spline bases in [36]. In an hp -d sense, we add an overlay of three B-splines of contracted knot span width to the original B-spline basis. At this point, no changes in the original basis functions are required, since we can infer from Equation (10) that single B-splines of contracted knot span width are linearly independent to original B-splines of full knot span width. The resulting basis is the combination of B-splines shown in Figure 7b, where original and overlay basis functions are plotted on separate levels, which reflects the two-level hierarchy between the original basis and its refinement overlay. Furthermore, we do not change the amplitude of contracted B-splines, thus ignoring the presence of scaling factors in Equation (10). We show later on that this simplification can be maintained, when we generalize this refinement strategy to higher dimensions and NURBS.

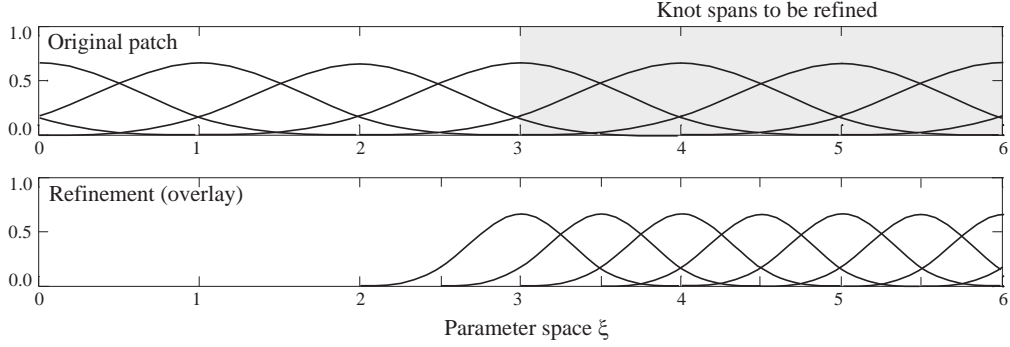


Figure 8: Hierarchical refinement in the sense of the hp -d adaptive approach for the three rightmost knot span elements in a row. The overlay is generated by repeating the nucleus operation of Figure 7b.

3.2.3. Two-level hierarchical refinement for several elements

The refinement rule introduced in Figure 7b for one element adds an overlay consisting of the contracted B-spline central to the element at issue as well as its right and left neighbor. Let us proceed one step further to the refinement of several knot span elements in a row. Figure 8 illustrates the two-level hierarchical basis, which results from a repetition of the nucleus operation illustrated in Figure 7b for the three rightmost elements in the patch. We do not add contracted B-splines a second time, if already generated from the refinement operation of a neighboring element. In particular, this procedure does not affect the higher-order continuity of the refined basis, since the first $p - 1$ derivatives of the hierarchical B-spline basis functions are zero at their support boundaries.

The reason for the specific choice of the nucleus operation becomes clear by consideration of the number of contracted B-splines. If we take fewer B-splines for each element refinement than shown in Figure 7b and omit the left and right neighbor, we would not obtain a complete row of contracted B-splines in Figure 8, since every second contracted B-spline would be missing. If we take more, the refinement would again spread out further beyond the leftmost element at issue, which would counteract the localization of refinement. The refinement rule of Figure 7b is valid for polynomial degree $p=3$, but can be easily transferred to B-spline bases of other polynomial degrees by looking for the minimum number of contracted B-splines per element, with which a complete row of contracted B-splines in the overlay level can be achieved, when several elements are refined.

3.2.4. Multi-level hierarchical refinement

In order to increase the degree of local refinement, we can repeat the procedure described in the previous paragraphs several times. In doing so, we proceed from the two-level hierarchy of a single refinement step to a general multi-level hierarchy, consisting of several overlay levels. Let us introduce the level counter k , where $k=0$ denotes the original B-spline patch. In each refinement step, the nucleus operation is applied to elements of the currently finest level k to produce a new

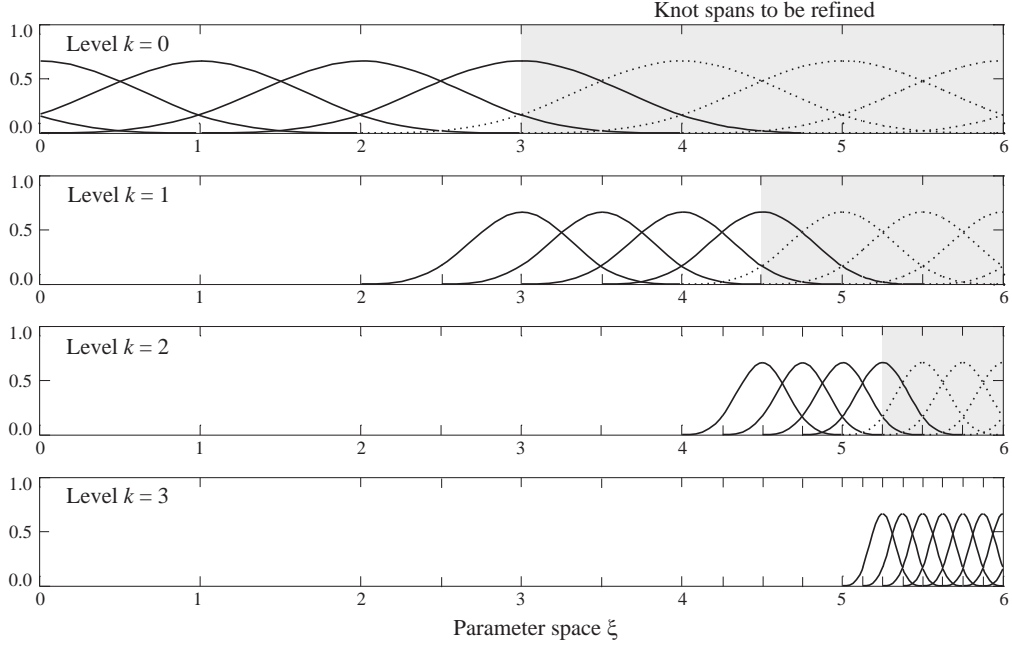


Figure 9: Hierarchical multi-level refinement: B-splines of level k plotted in dotted line can be represented by a linear combination of B-splines of the next level $k+1$ according to the two-scale relation Equation (10) and therefore need to be removed from the basis.

overlay level $k+1$. Hierarchically contracted B-splines of the new level $k+1$ are found by bisecting the knot span width with respect to level k , so that the specific width h_k of each level can be found by the relation

$$h_k = 2^{-k} h, \quad 1 \leq k \leq m \quad (13)$$

where h denotes the original knot span width of the unrefined B-spline patch and m the total number of levels in the hierarchy. The multi-level refinement procedure is illustrated in Figure 9, where the nucleus operation is successively applied to the three rightmost knot span elements of each level k . The resulting grid consists of a nested sequence of bisected knot span elements, and multiple hierarchical overlay levels of repeatedly contracted uniform B-splines.

3.2.5. Recovering linear independence

In order to guarantee full analysis suitability of the hierarchically refined B-spline basis, we have to ensure its linear independence. Comparing the different levels in the hierarchy of Figure 9, one can immediately observe that each overlay level $k+1$ consists of more than $p+2$ consecutive refined B-splines. As a consequence, their linear combination is capable of representing some of the B-spline basis functions of the previous level k according to the two-scale relation Equation (10). Therefore, we need to identify all B-spline basis functions that are a combination of refined B-spline basis functions of the next level $k+1$ and remove them from the hierarchical basis. In

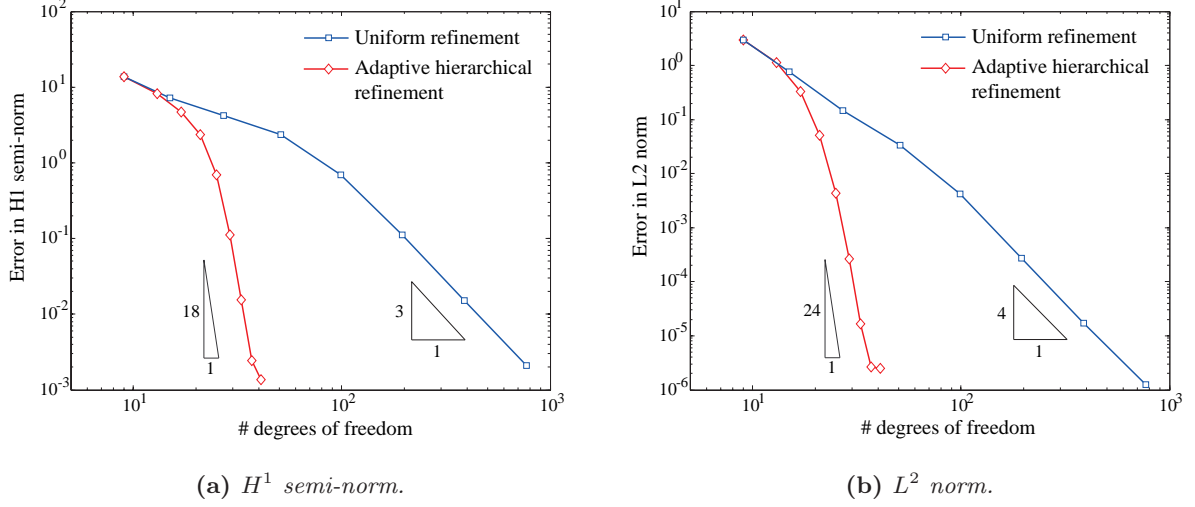


Figure 10: Convergence of cubic hierarchical B-spline bases for the 1D advection-diffusion example.

Figure 9, basis functions to be taken out are shown as dotted lines, while the final linear independent hierarchical B-spline basis consists of all basis functions shown as solid lines. The removal of linearly dependent basis functions improves the conditioning and the sparsity of the corresponding stiffness matrix, since the coupling of more contracted to less contracted B-splines through the hierarchy is considerably reduced. This can be observed in Figure 9, where basis functions of the lowest and the highest levels are completely decoupled, since they have no overlapping support.

3.3. A simple model problem in 1D

We test the efficiency of the hierarchically refined B-spline basis with a standard steady advection-diffusion problem in 1D, governed by the following equation and Dirichlet constraints

$$a \frac{\partial u}{\partial x} + D \frac{\partial^2 u}{\partial x^2} = 0 \quad (14a)$$

$$u(x=0) = 0; \quad u(x=L) = 1 \quad (14b)$$

Parameters $a = 100$, $D = 1$ and $L = 3$ denote the velocity, the diffusion coefficient and the length of the domain, respectively, and u is the unknown concentration. Dirichlet constraints are specified at both ends. Here, the nature of the problem is dominated by advection, indicated by the high global Péclet number $Pe = aL/D = 300$. This leads to a boundary layer at the right hand end of the 1D domain, which involves very high gradients in the solution. An in-depth discussion of this problem and its exact solution can be found in [73, 74].

We apply a standard Galerkin discretization, where the dominance of the non-symmetric advection operator over the diffusion operator in Equation (14a) leads to spurious oscillations. While these issues are usually addressed by consistent stabilization techniques [74, 75], we use a sequence

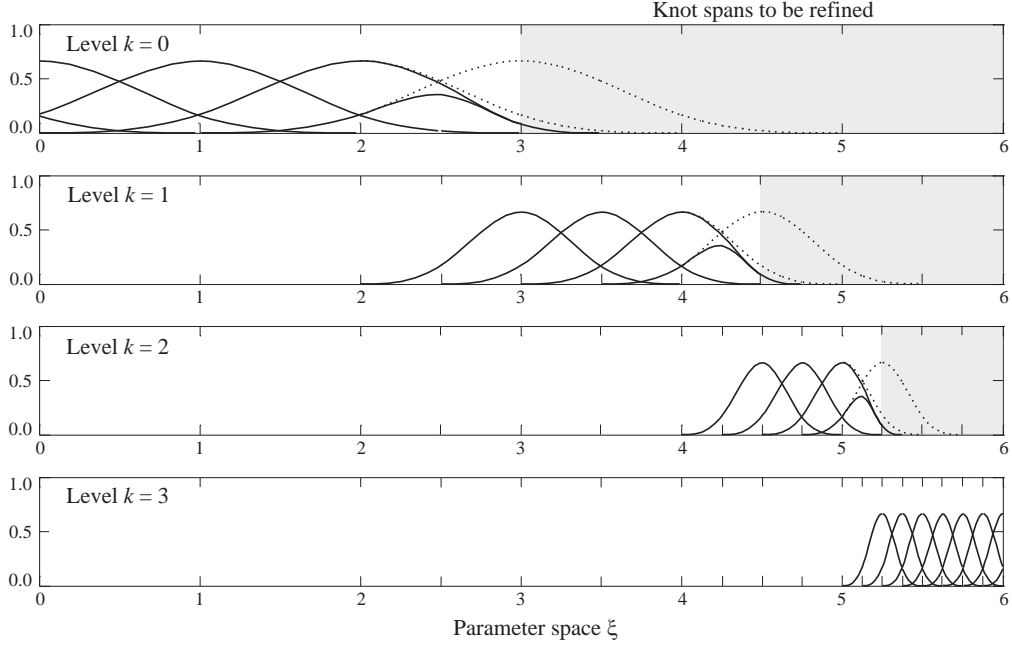


Figure 11: The subtraction procedure removes multiplicities of hierarchical B-splines according to the two-scale relation, which further reduces the coupling of B-splines throughout the hierarchy.

of cubic hierarchical bases in the sense of Figure 9 to obtain an accurate solution. In each refinement step, we generate an additional overlay level by adaptively refining the three rightmost elements. Since they are not interpolatory at the ends, the hierarchical basis requires weak enforcement of Dirichlet constraints, for which a simple penalty method [76, 77] with penalty parameter $\beta = 10^6$ is applied here. Figures 10a and 10b show the convergence in the H^1 semi-norm and L^2 norm, respectively, obtained with uniform h -refinement and adaptive hierarchical refinement. Uniform refinement doubles the number of equidistant knot span elements in each refinement step, which leads to optimal rates of convergence. Due to their adaptivity, the hierarchical bases achieve rates of convergence, which are far higher. To arrive at the final error level in both the H^1 and L^2 cases, the hierarchical bases require about one order of magnitude fewer degrees of freedom than uniform h -refinement. After seven refinement steps, the largest part of the error does not stem from the excessively refined right boundary anymore, so that the convergence rate levels off.

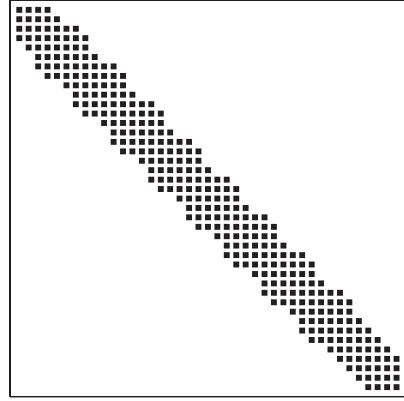
3.4. Condition number and sparsity of the stiffness matrix

In the hierarchically refined basis of Figure 9, some basis functions occur explicitly as refinement functions on some level k , but are also contained implicitly in some B-spline basis function of higher level according to the two-scale relation Equation (10). We can achieve a further improvement of the conditioning and sparsity of the stiffness matrix, if we detect and remove these multiplicities. In principle, this can be achieved by checking each hierarchical B-spline of the next level $k+1$ to



Number of dofs: 41
Number of non-zero entries: 399
Maximum bandwidth: 17

(a) *Without subtraction procedure.*



Number of dofs: 41
Number of non-zero entries: 307
Maximum bandwidth: 9

(b) *With subtraction procedure.*

Figure 12: *Sparsity pattern of a stiffness matrix without and with subtraction procedure.*

determine if it has common support with a B-spline of the current level k . If it does, we check whether the former is part of the linear combination of subdivision B-splines that result from the two-scale relation Equation (10) applied to the latter. If this is also true, we subtract the hierarchical B-spline of the next level $k+1$, multiplied by the corresponding scaling factor, from the B-spline of the current level k . Applying this subtraction procedure to the example basis of Figure 9, we arrive at the improved hierarchical basis of Figure 11, where initial basis functions are plotted as dotted lines, while the final basis functions after subtraction are plotted as solid lines. We observe that the overlapping of hierarchical levels is further reduced.

Analogous to the previous sub-section, we create two sequences of hierarchically refined bases with and without subtraction procedure, respectively, which are based on a refinement of the last three elements of each hierarchical level, and apply them to the advection-diffusion problem of Equation (14). To exclude any effect from weak boundary conditions, we replace uniform boundary basis functions by interpolatory basis functions created from open knot vectors, so that Dirichlet constraints can be satisfied strongly. The sparsity pattern of the stiffness matrices without and with subtraction procedure and the evolution of the corresponding condition numbers with increasing levels of hierarchical refinement are illustrated in Figure 12 and Table I, respectively.

# of hierarchical levels k	0	1	2	3	4	5	6	7	8
Without subtraction	42.0	84.8	85.6	84.5	88.3	106.3	197.6	388.6	774.1
With subtraction	42.0	51.2	44.2	41.4	44.8	67.0	126.3	250.8	502.3

Table 1: *Condition number of the stiffness matrix for different numbers of hierarchical levels without and with subtraction procedure.*

The subtraction procedure has a beneficial effect on the structure of the sparse matrix, reducing both bandwidth and number of matrix entries. Note that the ordering of matrices was optimized by the symmetric reverse Cuthill-McKee algorithm [78]. The condition number of the matrix is improved by the subtraction of multiplicities, but still remains in the order of magnitude of its counterpart without subtraction in the one-dimensional case. We note that in higher dimensions, the difference in the condition number might be more pronounced, since there are many more basis functions that have common support and belong to different hierarchical levels.

From an implementation point of view, the detection of multiplicities requires complex algorithms, which obstruct automation of the refinement process. Its advantages are likely to be moderate, since the performance of iterative solvers is dominated by the condition number of the matrix, while the bandwidth plays a subordinate role. Therefore, we omit the subtraction procedure in the following, when we will focus on efficient and easy-to-implement hierarchical refinement schemes for isogeometric analysis in multiple dimensions.

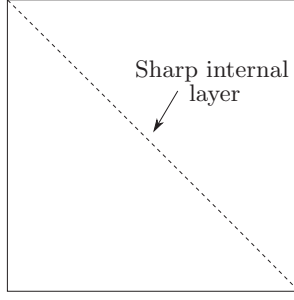
4. Hierarchical refinement of B-splines in multiple dimensions

Due to their tensor-product structure, the concept of hierarchical refinement in 1D directly carries over to multivariate B-splines. We discuss its implementation in the framework of tree-like data structures and suggest corresponding algorithms. To keep things simple at this stage, we confine ourselves to axis-aligned B-spline discretizations of cuboidal domains.

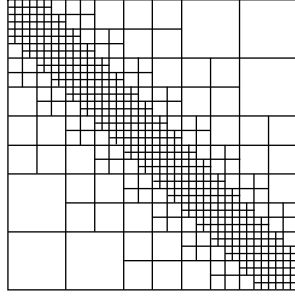
4.1. Transition from the 1D concept to multiple dimensions

The tensor product structure of multivariate B-splines permits a straightforward generalization of the one-dimensional hierarchical refinement concept presented in Section 3 to multiple dimensions. Assume a d_p -dimensional B-spline patch, which is generated according to Section 2.2 by d_p univariate knot vectors in each parametric direction ξ^ℓ . An adaptive multi-level B-spline basis can be generated by successively applying the 1D procedures of the previous section, since the tensor product structure allows for a decoupling of refinement operations in each parametric direction ℓ and subsequent d_p -dimensional assembly by multiplication. We will illustrate this by commenting briefly on each refinement step:

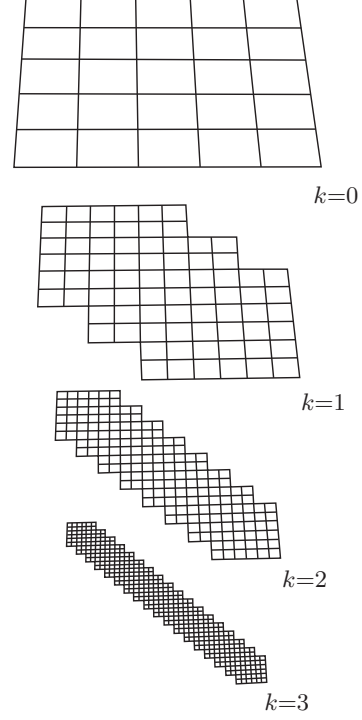
- *Nucleus operation* (refinement of a single d_p -dimensional element): Depending on the polynomial degree p_ℓ of each parametric direction, we choose a suitable number of contracted B-splines for each direction ℓ according to the principles outlined in Section 3.2.2. Assuming $p = 3$, the nucleus operation in each parametric direction ℓ corresponds to Figure 7b, from which d_p -dimensional hierarchical B-splines can be generated by multiplying the one-dimensional functions in the sense of Figure 6.



(a) *System sketch.*



(b) *Hierarchical mesh.*



(c) *Multi-level structure of hierarchically contracted knot spans.*

Figure 13: Multi-level hierarchical B-splines of $p=3$ for adaptive refinement along an internal layer in a square domain. The original patch of level $k=0$ consists of 5×5 knot span elements.

- *Multi-level hierarchy:* The local increase of refinement follows the same concept as described in Sections 3.2.3 and 3.2.4, where the repetition of the nucleus operation in each overlay level k and the repetition of hierarchical refinement with successively contracted B-splines for the generation of the next overlay level $k+1$ lead to a multi-level B-spline basis, which naturally accommodates adaptivity in d_p dimensions.
- *Recovering linear independence:* A linear combination of multivariate hierarchical B-splines of the next level $k+1$ can represent B-splines of the current level k in the sense of the multivariate two-scale relation Equation (12). These linear dependencies need to be identified through the multi-level hierarchy and eliminated by a removal of corresponding higher-level B-splines. In analogy with Section 3.2.5, this can be achieved by determining in each parametric direction if the required $p+2$ contracted B-splines of level $k+1$ exist in the hierarchical structure.
- *Dirichlet constraints:* Dirichlet boundary conditions can be incorporated weakly by variational methods [77, 79–82] or strongly by a least squares fit of boundary basis functions [5, 83]. Homogeneous boundary conditions can be imposed strongly by removing all basis

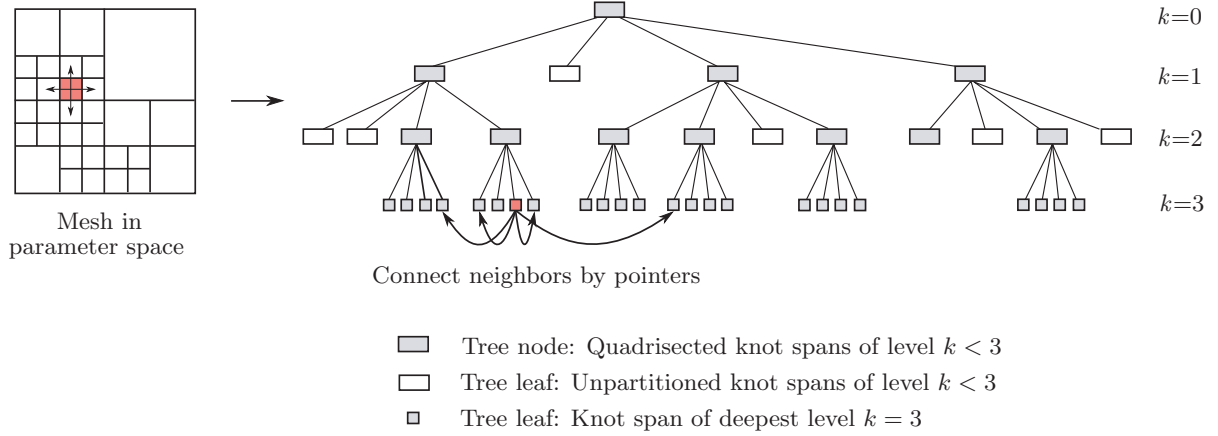


Figure 14: Quadtree example illustrating the hierarchical data organization of part of an adaptive mesh. The neighboring relations within each hierarchical level are established by pointers, which are shown here for one element of the finest level (in red color).

functions with support at the Dirichlet boundary from the basis.

Multivariate hierarchical refinement is illustrated for the example of a 2D square domain, which should be refined along its diagonal as shown in Figure 13a. Figure 13b shows the hierarchical mesh, which represents the element structure, over which numerical integration is carried out, so that the coupling of basis functions from different levels can be taken into account. Basis functions of different levels are defined over knot spans, which are hierarchically contracted in the sense of Equation (13). The corresponding multi-level overlay structure is illustrated in Figure 13c, where the sequence of hierarchical knot spans is plotted.

4.2. Geometry representation and hierarchical mapping

In the general case, tensor-product B-splines are mapped from a regular grid with respect to the parameter space to an arbitrarily shaped physical geometry via control point values according to Equation (6). In the special case of an axis-aligned regular B-spline mesh, we can use the simplicity of its cuboidal geometry to describe the mapping from parameter space ξ to physical space x analytically by the following transformation

$$x = x_0(\xi) + H \xi \quad (15)$$

where x_0 denotes the origin of the physical domain in the parameter space, and H is a diagonal matrix, containing the physical knot span width $h_{\xi\ell}$ of each parametric direction ℓ . The Jacobian determinant directly follows as

$$J = \det H \quad (16)$$

Due to Equations (15) and (16), the B-spline basis can be disconnected completely from the geometry and serves exclusively for the approximation of the solution fields.

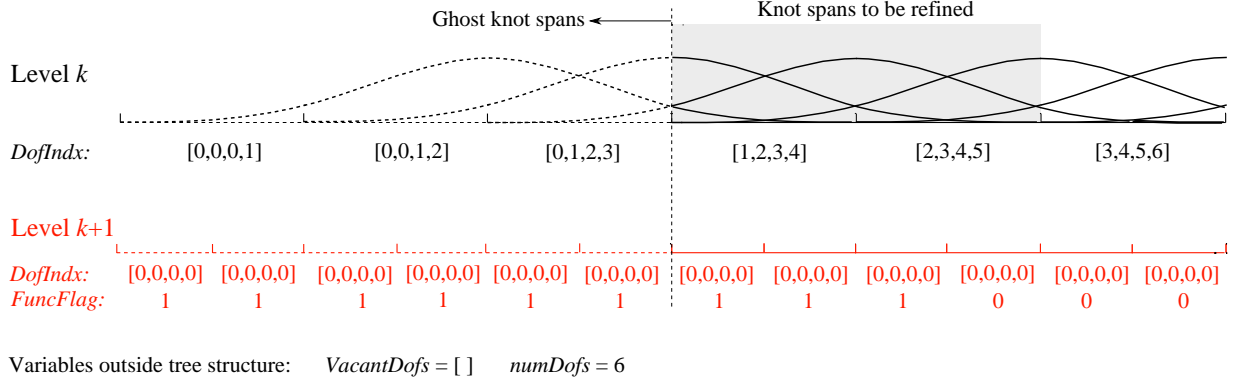


Figure 15: Illustration of Algorithm 1: Apply nucleus operation to the first two knot span elements, which creates new knot spans of level $k+1$ and flag knot spans, where contracted basis functions start. Also split and flag all related ghost knot spans in order to ensure a correct treatment of boundary basis functions.

Numerical integration is performed over the hierarchical mesh (see Figure 13b). Depending on the hierarchical level k of the respective element, an additional mapping with Jacobian

$$j = \left(\frac{1}{2^k} \right)^{d_p} \quad (17)$$

is required, where d_p is the number of dimensions of the B-spline discretization. In the scope of the present paper, integrals are evaluated by Gauss integration, which places $(p+1)^{d_p}$ integration points in each element of the hierarchical mesh.

4.3. Efficient implementation in tree data structures

A considerable advantage of hierarchical refinement in the present form is the preservation of its intuitive multi-level concept irrespective of the dimensionality of the B-spline basis, which allows for an easy generalization of corresponding algorithms in the framework of the tensor product structure. Additionally, its direct correspondence to standard multi-level data structures [40] constitutes the basis for its straightforward and efficient implementation.

4.3.1. Quad- and octrees

Hierarchical data structures provide a natural way to decompose and organize spatial data according to different levels of complexity and offer fast access to relevant parts of a dataset [40]. For their efficient implementation, binary, quad- and octrees are usually employed in one, two and three dimensions, respectively [41, 42, 84]. The quadtree concept shown in Figure 14 illustrates the analogy between an adaptive hierarchical quadrilateral mesh and the two-dimensional tree. In our implementation of hierarchical B-spline refinement, the tree is the fundamental entity, where each node or leaf holds all the information of the corresponding knot span on the respective hierarchical level and of the basis functions defined therein. Additionally, we equip each node or leaf with pointers that connect it with all direct neighbors of the same hierarchical level (see Figure 14). These

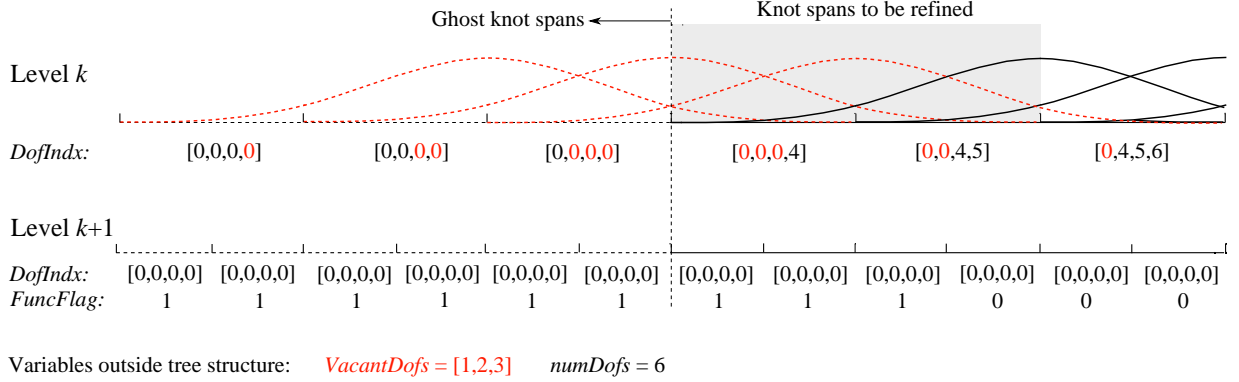


Figure 16: Illustration of Algorithm 2: Identify linear dependencies, remove corresponding degrees of freedom and store them in *VacantDofs* for later reassignment.

“horizontal” neighboring relations are frequently needed during refinement to establish contracted basis functions, check for linear dependencies and correctly assign degrees of freedom.

4.3.2. Evaluation of basis functions

Hierarchical B-splines N^k of knot span width $h/2^k$ are generated by contraction of unrefined B-splines. Thus, they do not have to be implemented separately, but can be directly computed on each overlay level k in each parametric direction ℓ from their original unrefined counterparts N^0 as follows

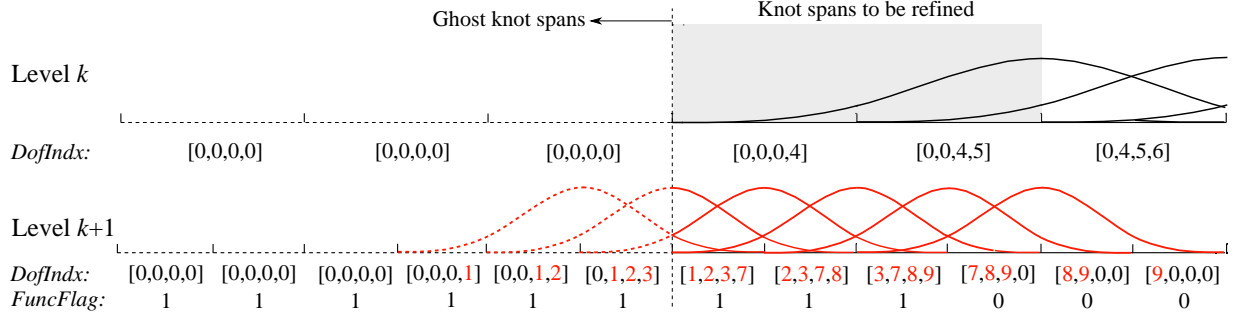
$$N^k(\xi^\ell) = N^0(2^k \xi^\ell) \quad (18a)$$

$$\frac{\partial N^k(\xi^\ell)}{\partial \xi^\ell} = \frac{1}{2^k} \frac{\partial N^0(2^k \xi^\ell)}{\partial \xi^\ell} \quad (18b)$$

According to their tensor-product structure, multivariate B-splines can be subsequently assembled by simple multiplication of their components from each parametric direction.

4.3.3. Degree of freedom organization

Handling degrees of freedom within the hierarchical tree structure is complicated by the removal of basis functions during the refinement process and therefore requires special care. The degrees of freedom attributed to the basis functions with support in a knot span element are contained in an array structure, denoted as *DofIdx* here, and stored in the corresponding node or leaf of the tree. In particular, the last entry of *DofIdx* corresponds to the B-spline whose support starts in the current knot span and continues over the successive $p+1$ knot spans in positive parametric direction. A zero in *DofIdx* indicates that the corresponding basis function does not exist in the basis. This idea can be easily generalized to d_p dimensions, where the support of the B-spline that starts in the current d_p -dimensional knot span continues over a regular polytope spanned by the successive $(p+1)^{d_p}$ knot spans (a line in 1D, a square in 2D, a cube in 3D, etc.). A 1D example of this numbering concept, which is similar to the one devised in Höllig [35], is given in Figure 15.



Variables outside tree structure: $VacantDofs = []$ $numDofs = 9$

Figure 17: Illustration of Algorithm 3: On the basis of *FuncFlag*, add contracted basis functions of level $k+1$ that have support outside the ghost knot spans. Assign degrees of freedom by filling *DofIdx* in positive parameter direction. Consume degrees of freedom buffered in *VacantDofs* first.

During refinement, contracted B-splines of the next hierarchical level $k+1$ are first initiated via a Boolean, called *FuncFlag* here, to flag the respective leaf, where the new basis function starts. This allows us to check for linear dependencies first, to remove corresponding B-splines of level k by setting their *DofIdx* entries to zero and to buffer their degree of freedom numbers in the array *VacantDofs*. Later on, we can reassign these numbers to basis functions of the new hierarchical level $k+1$, until the buffer *VacantDofs* is empty. In order to carry out the same operations on boundary knot spans, we introduce ghost knot spans [42, 85] that are taken into account only during refinement, but not during analysis (see Figures 15 through 17).

4.3.4. Basic algorithmic sketch

We roughly outline the basic algorithmic ideas of a single hierarchical refinement step. As input, we assume the tree structure of the current hierarchical level k and the result of an error estimator, which specifies for each leaf of the tree if it is to be refined or not. For simplicity, we consider here only the leaves of the currently finest level k . The refinement procedure consists of three parts. The first part is outlined in pseudocode in Algorithm 1 and illustrated in Figure 15 by a small example. It carries out the nucleus operation for each element to be refined, creates new leaves of level $k+1$ and fills in corresponding contracted basis functions via *FuncFlag*. The second part is given in pseudocode in Algorithm 2 and illustrated in Figure 16. It removes linear dependencies between basis functions of level k and $k+1$. The third part is outlined in pseudocode in Algorithm 3 and illustrated in Figure 17. It assigns degrees of freedom for the new basis functions of level $k+1$. As output, we receive the refined B-spline patch, organized in a tree structure of level $k+1$.

5. Hierarchical refinement of NURBS

Up to this point, we have dealt with B-splines over structured grids for the discretization of axis-aligned cuboidal domains. The concept of subdivision can also be applied to NURBS bases,

Data: Tree data structure, deepest level of leaves is k ; result of the error estimator for each knot span element of level k ;

Result: Adds new leaves of level $k+1$ to the tree; initialize index structure $DofIdx$; mark all new leaves with $FuncFlag$, where a new basis function starts;

```
// Loop over all knot span elements of level k (currently deepest level);
// Arguments of for-loops are in the sense of iterators, pointing to leaves in the tree;
for i_ele_k ← 1 to n_ele_k do
    if error estimator requires refinement in i_ele_k then
        // Apply nucleus operation to current i_ele_k;
        // Loop over all elements affected: In case of p=3, these are the current element
        // i_ele_k and its surrounding direct neighbors (see Figure 7b);
        // if i_ele_k is located at the boundary, also refine all neighboring ghost elements;
        for j_ele ← 1 to n_ele_affected do
            // Create new knot span elements of level k+1;
            if element j_ele is unrefined then
                Append 2/4/8 new leaves of level k+1 in the tree structure
                (bi-, quadri- or octasection of element in 1/2/3D, respectively);
                Initialize in each new leaf  $DofIdx = [0]$ ;  $FuncFlag = false$ ;
                Connect neighboring leaves of level k+1 by pointers and update pointers in
                all surrounding other leaves, if existing;
            end
            // Loop over all new leaves of j_ele;
            for i_leaf_k+1 ← 1 to n_leaves do
                // Fill in hierarchical basis functions of level k+1, if required by the
                // nucleus operation (see Figure 7b and Section 3.2.3);
                if a basis function starts in i_leaf_k+1 then
                    | Set i_leaf_k+1.FuncFlag = true;
                end
                // Ensure the proper handling of boundary basis functions;
                if i_leaf_k+1 contains a ghost element then
                    | Set i_leaf_k+1.FuncFlag = true;
                end
            end
        end
    end
end
end
```

Algorithm 1: Find elements and basis functions of the next hierarchical level $k+1$

which discretize arbitrarily shaped domains. A hierarchical refinement scheme for adaptive analysis with NURBS can be derived with minimal effort on the basis of the algorithms and data structures described in Section 4.3.

5.1. Refinability of NURBS basis functions by subdivision

Subdivision rules for univariate NURBS are derived by inserting the two-scale relation of Equation (10) into the construction rule for NURBS basis functions, Equation (7), which yields

$$R_{i,p}(\xi) = \frac{w_i 2^{-p} \sum_{j=0}^{p+1} \binom{p+1}{j} N_{i,p}(2\xi - j)}{\sum_{j=1}^n w_j N_{j,p}(\xi)} \quad (19)$$

```

Data: Tree data structure, deepest level of leaves is  $k+1$ ;
Result: Degrees of freedom of linearly dependent basis functions of level  $k$  are erased from
           $DofIndx$  in the tree and stored in  $VacantDofs$  for later reassignment;

// Loop again over all knot span elements of level  $k$ ;
for  $i\_ele\_k \leftarrow 1$  to  $n\_ele\_k$  do
    if  $i\_ele\_k$  has leaves of level  $k+1$  then
        // Implementation sketch of conditional clauses:
        // - start from the leaves of level  $k+1$  of the current element  $i\_ele\_k$ ;
        // - regular polytope at issue consists of  $(p+2)^d$  successive leaves of level  $k+1$ ;
        // - it is spanned by  $(p+2)$  leaves of level  $k+1$  in  $d$  positive parametric directions;
        // - walk through the polytope using pointers that connect neighboring leaves;
        // - determine existence of neighboring element by checking corresponding pointer;
        // Note that due to the ghost element concept (see Figures 15 through 17), this
        // strategy also works for boundary basis functions;
        if at least one of the leaves of the polytope does not exist then break;
        if  $FuncFlag == false$  in at least one of the leaves of the polytope then break;
        // Since in each leaf of the polytope, a basis functions of level  $k+1$  starts, the
        // basis function of level  $k$  starting in  $i\_ele\_k$  has to be removed and its degree
        // of freedom needs to be remembered for reassignment later on;
        Identify degree of freedom  $dof$  that starts in  $i\_leaf\_k$  and append to  $VacantDofs$ 
        // Loop over polytope spanned by  $(p+1)^d$  leaves of level  $k$  and erase  $dof$ ;
        for  $j\_ele\_k \leftarrow 1$  to  $n\_ele\_polytope$  do
            | Set  $j\_ele\_k.DofIndx$  ( position of  $dof$  ) = 0;
        end
    end
end
end

```

Algorithm 2: Remove linear dependencies between current level k and next level $k+1$

According to the isogeometric paradigm [1, 5], the geometry is described exactly by the original unrefined NURBS basis, so that geometry refinement in the framework of isogeometric analysis is not required. Therefore, we keep the weights w_j and corresponding control points \mathbf{P}_j unchanged and always take the sum of the original B-splines $N_{j,p}$ in the denominator of Equation (19). Nonetheless, using the refined NURBS basis for enhancing the geometry representation would be of course possible [86, 87].

A multivariate subdivision rule for NURBS can be derived analogously by substituting Equation (12) into Equation (8)

$$R_{i,p}^h(\xi) = \frac{w_i \sum_j \left(\prod_{\ell=1}^d 2^{-p_\ell} \binom{p_\ell+1}{j_\ell} N_{p_\ell}(2\xi^\ell - j_\ell) \right)}{\sum_j w_j B_{j,p}(\xi)} \quad (20)$$

Data: Tree data structure, deepest level of leaves is $k+1$; list of unassigned degrees of freedom in *VacantDofs*; total number of degrees of freedom *numDofs*;
Result: In leaves of level $k+1$ with *FuncFlag*==true, a degree of freedom is assigned to the basis function starting there;

```
// Loop over all new knot span elements of level k+1;
for i_leaf_k+1 ← 1 to n_leaves_k+1 do
    // Prevent dof assignment to basis functions that consist of ghost knot spans only;
    // Consider polytope spanned by (p+1)d leaves of level k+1;
    if all leaves of the polytope are ghost knot spans then break;
    else if i_leaf_k+1.FuncFlag == true then
        if VacantDofs is empty then
            // Loop over polytope spanned by (p+1)d leaves of level k+1 and assign new dof;
            for j_leaf_k ← 1 to n_polytope do
                Set i_leaf_k+1.DofIdx ( position of numDofs ) = numDofs;
            end
            numDofs++;
        else
            // Assign degree of freedom from VacantDofs;
            for j_leaf_k ← 1 to n_polytope do
                Set i_leaf_k+1.DofIdx ( position of numDofs ) = VacantDofs.First();
            end
            Erase first entry of VacantDofs;
        end
    end
end
end
```

Algorithm 3: Activate new basis functions

where the multi-index notation exactly follows the one introduced in Section 2.2 for multivariate B-splines and Section 3.1 for multivariate B-spline subdivision. Since the geometry is not refined, B-splines $B_{\mathbf{j},\mathbf{p}}$ and corresponding weights in the denominator of Equation (20) refer again to the original B-spline patch.

5.2. Hierarchical refinement of NURBS

On the basis of Section 5.1, we introduce some adaptations to the hierarchical refinement scheme for B-splines to also accommodate NURBS. First, we separate Equations (19) and (20) in a B-spline part (numerator) and a rational part (denominator), which are treated separately. The numerator carries out hierarchical refinement on the B-spline level, so that we can make full use of the concepts and algorithms introduced in Sections 3 and 4. Thus, the resulting refined NURBS basis is also constructed from a nested sequence of bisected knot span elements, over which multiple hierarchical overlay levels of repeatedly contracted B-splines are defined.

As shown in the previous sub-section, the denominator is always computed with the original B-spline basis $B_{\mathbf{j},\mathbf{p}}^0(\boldsymbol{\xi})$

$$\text{sum}(\boldsymbol{\xi}) = \sum_{\mathbf{j}} w_{\mathbf{j}} B_{\mathbf{j},\mathbf{p}}^0(\boldsymbol{\xi}) \quad (21)$$

where multi-index \mathbf{j} includes all B-splines with support at the parameter space location $\boldsymbol{\xi}$. The basis functions $R_{\mathbf{i}}$ of the hierarchical NURBS basis and its derivatives follow as

$$R_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) = \frac{B_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi})}{\text{sum}(\boldsymbol{\xi})} \quad (22a)$$

$$\frac{\partial R_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi})}{\partial \xi^\ell} = \frac{\partial B_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) / \partial \xi^\ell \text{sum}(\boldsymbol{\xi}) - B_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) \partial \text{sum}(\boldsymbol{\xi}) / \partial \xi^\ell}{\text{sum}(\boldsymbol{\xi})^2} \quad (22b)$$

where we additionally drop the weights $w_{\mathbf{i}}$ in the numerator of Equation (22a) for further simplification. Standard rules for generating higher order derivatives [63] can be adapted in the same way. The geometry mapping by way of the Jacobian matrix and determinant are computed throughout the hierarchical refinement procedure from the unrefined NURBS basis

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{\mathbf{j}} \frac{w_{\mathbf{j}} B_{\mathbf{j},\mathbf{p}}^0(\boldsymbol{\xi})}{\text{sum}(\boldsymbol{\xi})} \mathbf{P}_{\mathbf{j}} \quad (23)$$

with the initial set of weights $w_{\mathbf{j}}$ and control points $\mathbf{P}_{\mathbf{j}}$.

6. Numerical examples of adaptive isogeometric analysis

In the following, the versatility of hierarchical refinement in the framework of isogeometric analysis is demonstrated for a series of fluid and solid mechanics problems in two and three dimensions. B-spline and NURBS basis functions exhibiting higher order continuity have been shown to be an

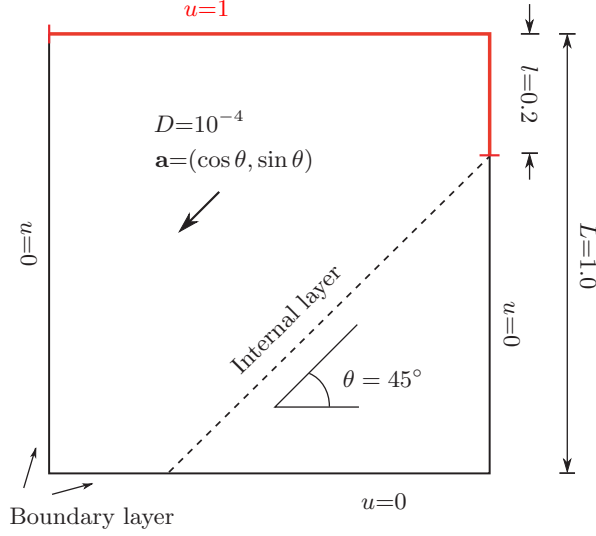


Figure 18: *Advection skew to the mesh in 2D: Problem definition.*

ideal candidate for approximating these problems in the framework of the finite element method [1, 5]. Therefore, we will not discuss their general solution characteristics, but directly concentrate on their hierarchical refinement. All examples are discretized with cubic B-splines or NURBS. For the solution of the linear system of equations, we use an iterative GMRES solver with ILU preconditioning provided by the library framework Trilinos [88].

6.1. Error estimation and automatic refinement

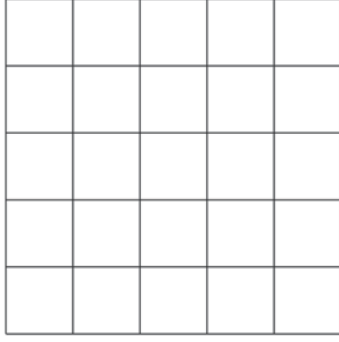
In the following, elements to be refined are selected automatically by means of a simple gradient based error indicator ε . Since we aim at capturing steep gradients in the solution, we use

$$\varepsilon_e = \frac{1}{V_e} \left(\int_{\Omega_e} |\nabla \mathbf{u}|^2 \, d\Omega_e \right)^{1/2} \quad (24)$$

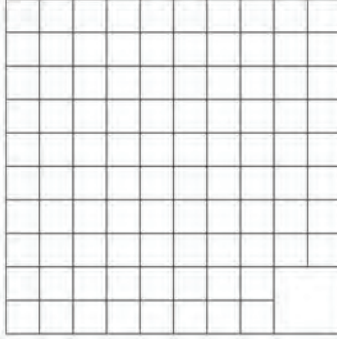
where \mathbf{u} is the solution field evaluated from the current mesh of hierarchical level k . The indicator is evaluated over the domain Ω_e of each element and subsequently normalized with respect to the corresponding element volume V_e . If ε_e is larger than its average

$$\varepsilon_e > \frac{C}{n_{ele}} \sum_{i=1}^{n_{ele}} \varepsilon_e^i \quad (25)$$

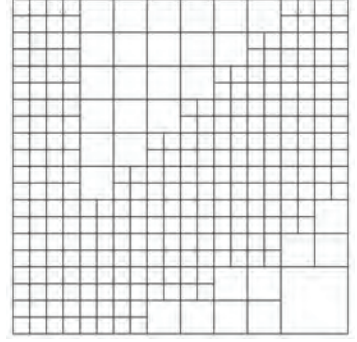
with n_{ele} being the number of all elements in the current mesh, the element is refined. We introduce an additional constant C to empirically fine-tune the threshold for the specific problem. For more elaborate error estimators, see for example [72].



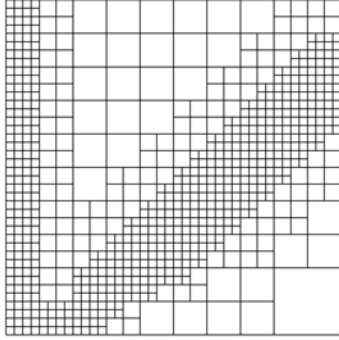
(a) Uniform mesh $k=0$.
 $n_{dof} = 49$



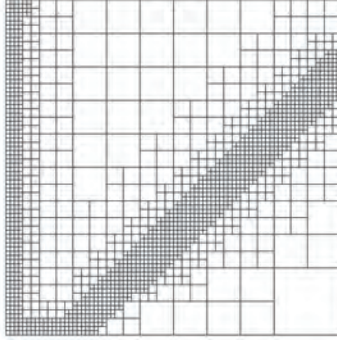
(b) Mesh of level $k=1$.
 $n_{dof} = 119$



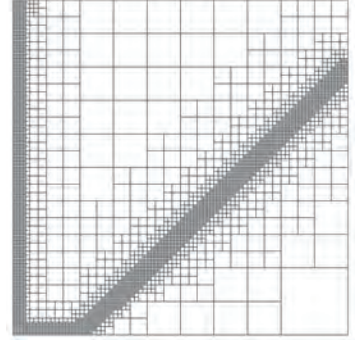
(c) Mesh of level $k=2$.
 $n_{dof} = 279$



(d) Mesh of level $k=3$.
 $n_{dof} = 612$



(e) Mesh of level $k=4$.
 $n_{dof} = 1,319$



(f) Mesh of level $k=5$.
 $n_{dof} = 3,017$

Figure 19: Sequence of hierarchical meshes generated by an automatic refinement scheme that makes use of a simple gradient-based error indicator and the algorithms described in Section 4.4.

6.2. Cuboidal B-spline geometries

We will first focus on problems over cuboidal axis-aligned domains, which can be discretized exactly by B-splines defined over a structured grid of knot span elements. As outlined in Section 4.2, this allows for a particularly simple geometry handling.

6.2.1. Advection skew to the mesh in 2D

The first model problem is illustrated in Figure 18a and involves the solution of the linear advection-diffusion equation

$$\mathbf{a} \cdot \mathbf{u} - \nabla \mathbf{u} \cdot (D \nabla \mathbf{u}) = f \quad (26)$$

where u denotes the solution, \mathbf{a} is the velocity, D is the diffusivity and f is a source term. In particular, the velocity is inclined to the mesh at 45° and the diffusivity is chosen extremely small, so that the problem is dominated by advection, resulting in a very high global Péclet number of $Pe = 10^4$. Thus, we expect sharp interior and boundary layers, which require stable numerical

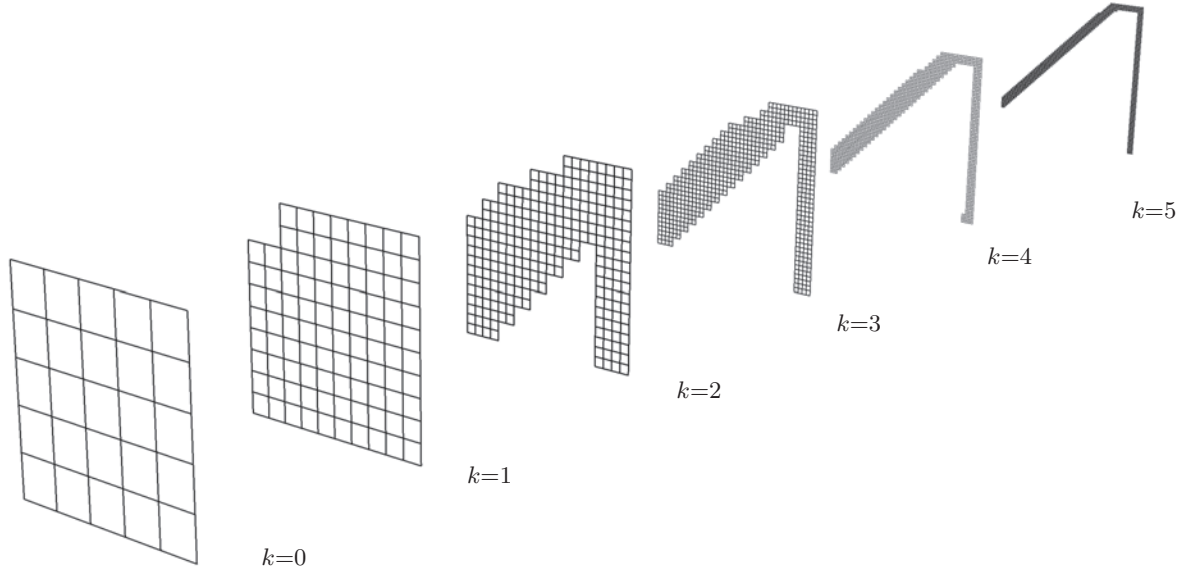
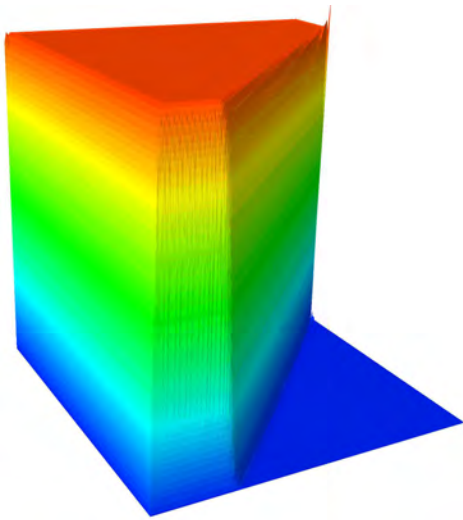


Figure 20: Sequence of knot span elements for each hierarchical level k , corresponding to the hierarchical meshes for the 2D advection problem.

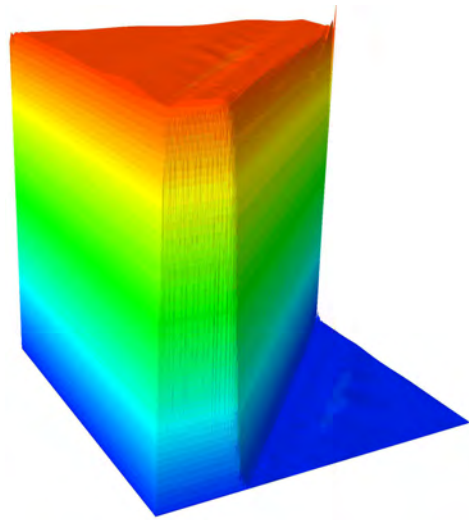
techniques in addition to increased resolution to be accurately captured. The problem is a well-studied benchmark [75, 79], examined for uniform k -refinement in [5] and local T-spline refinement in [89, 90].

We investigate the adaptive resolution of the internal and boundary layers with the present hierarchical refinement approach, starting from a 5×5 grid of cubic B-splines. We satisfy boundary conditions weakly at the inflow boundary with Nitsche’s method [80–82], where the penalty parameter β is 50, and strongly at the outflow boundaries [79]. Furthermore, we apply standard SUPG stabilization [74, 75, 79] in addition to refinement. The corresponding stabilization parameter is chosen to be $\tau = h_a / (2|\mathbf{a}|)$, where h_a is the element length of in flow direction. In the current example, $h_a = \sqrt{2} 2^{-k} h$, where k is the finest level of hierarchical refinement present in the element under consideration and h is the original knot span spacing of the unrefined discretization. Employing an automatic refinement scheme based on the error indicator of Equation (24) and the algorithms described in Section 4.3, we obtain a sequence of hierarchical meshes shown in Figure 19. The hierarchy of knot spans defining B-splines of each level k is shown in Figure 20. It can be observed that the refinement captures the location of the internal as well as the boundary layers very well.

An overkill solution obtained with a uniform cubic B-spline discretization of 160×160 elements and SUPG stabilization and the adaptive solution obtained with a refined mesh of hierarchical level $k = 5$ are shown in Figure 20. We can observe over- and under-shooting of the adaptive solution along the internal layer as also reported in [89, 90] for T-spline refinement. Nonetheless, the comparison of adaptive and uniform solution fields demonstrates that hierarchical refinement leads



(a) Solution obtained with a uniform discretization of 160×160 cubic elements.



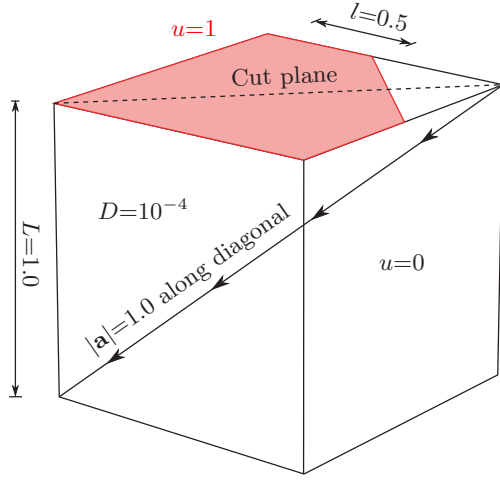
(b) Solution obtained with the adaptive hierarchical mesh of $k = 5$.

Figure 21: Solution fields of the advection dominated problem in 2D.

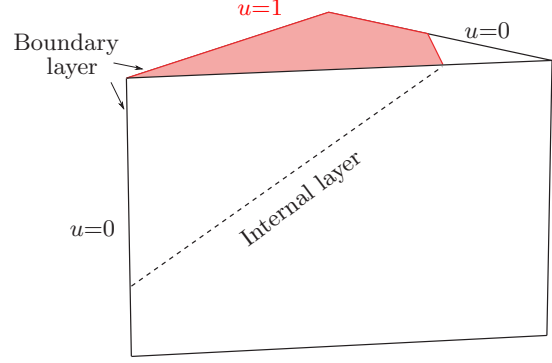
to a qualitatively similar result. While both meshes provide a comparable element size around the location of the layers, the adaptive mesh with 3,017 dofs requires only about 12% of the degrees of freedom of the uniform mesh with 26,244 dofs. Finally, we would like to point out the high quality of the refinement in terms of locality, as the hierarchical elements of the finest level show no propagation through the mesh. This is an improvement in comparison to recent works on adaptive isogeometric analysis [89, 90], where automatic local refinement employing cubic T-splines was reported to result in globally refined T-meshes when applied to a very similar advection-diffusion problem. However, in a recent work, an effective T-spline local refinement algorithm has been developed [30].

6.2.2. Advection skew to the mesh in 3D

We construct an analogous advection-diffusion problem in three dimensions, whose details are given in Figure 22a. It exhibits the same challenges in terms of strong advection dominance (global Péclet number $Pe = 10^4$), a sharp internal layer and several boundary layers (see Figure 22b). Following the previous sub-section, we apply SUPG stabilization and the error indicator of Equation (24). We also impose weak and strong boundary conditions at the inflow and outflow boundaries, respectively, where the penalty parameter β of Nitsche's method is set to 50 again. Figure 23 shows the resulting hierarchical mesh of level $k = 4$, generated automatically from an initial $6 \times 6 \times 6$ cubic B-spline grid, and the corresponding solution field. The mesh captures the location of internal and boundary layers accurately and its refinement is local without propagating throughout the entire domain. Similar to the 2D case, the solution field exhibits slight under- and overshooting along the internal layer. While the adaptive mesh of Figure 23a features only 116,314 degrees of



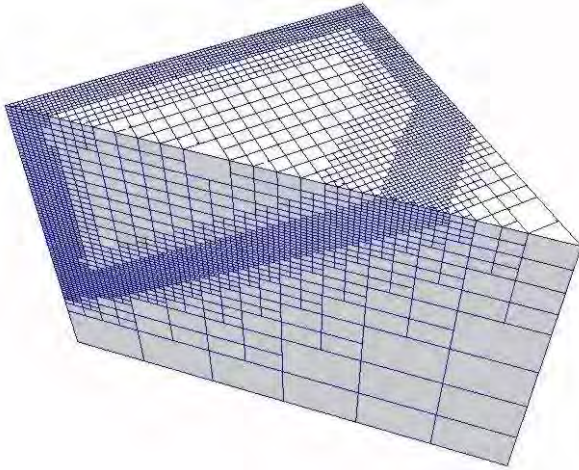
(a) Problem definition.



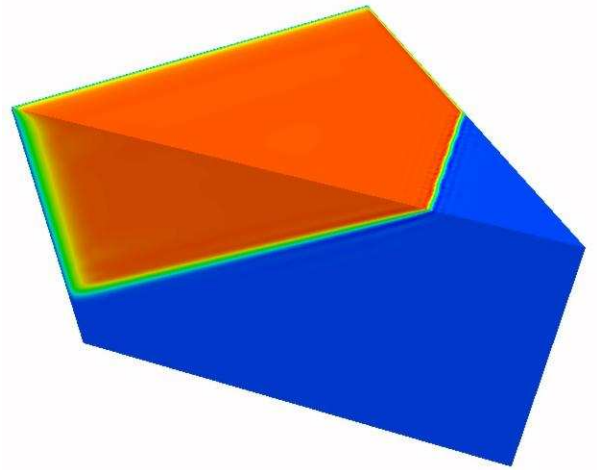
(b) System cut along diagonal plane.

Figure 22: Advection skew to the mesh in 3D.

freedom, a uniform cubic B-spline discretization, which uses a global mesh size corresponding to the finest elements of the adaptive mesh, requires a resolution of $96 \times 96 \times 96$ knot span elements, which amounts to 941,192 degrees of freedom. In Section 4, we shown that our hierarchical refinement approach can be generalized easily to three dimensions from algorithmic and implementation points of view. The present result confirms that hierarchical refinement in 3D also yields high quality adaptive meshes and accurate solution fields, comparable to the 2D case of the previous sub-section.



(a) Hierarchical mesh of level $k=4$.



(b) Accurate approximation of sharp layers.

Figure 23: Adaptive mesh and corresponding solution for the 3D advection problem. To reveal internal features, only one half of the domain is plotted according to Figure 23b.

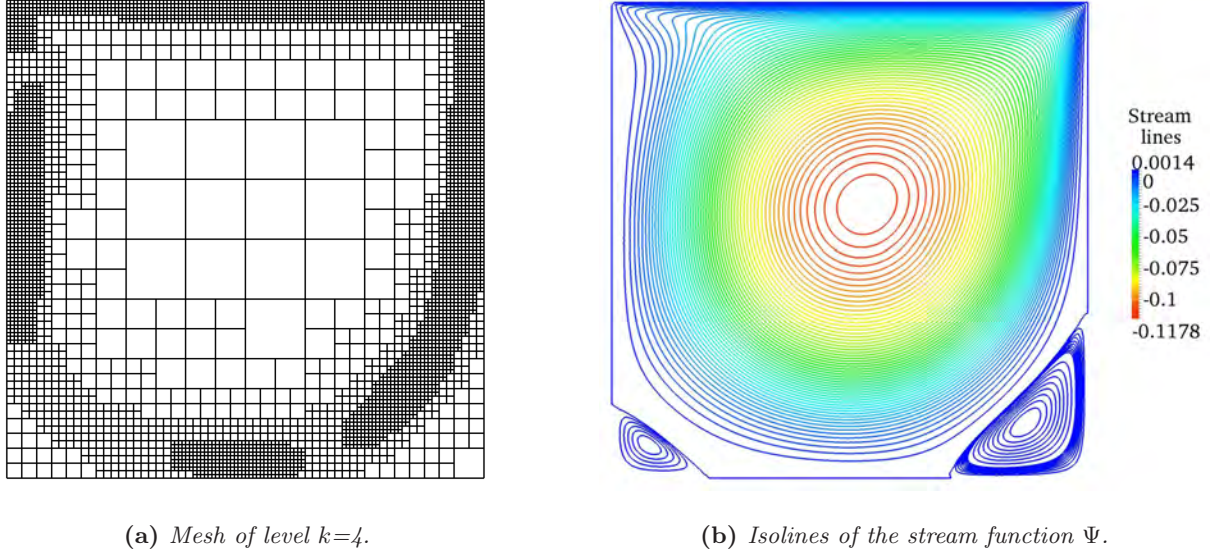


Figure 24: The lid driven cavity problem: Adaptive hierarchical mesh and corresponding solution field.

6.2.3. Lid driven cavity

To demonstrate the versatility of the hierarchical refinement approach in the framework of a further physical model, we consider the two-dimensional stationary incompressible Navier-Stokes equation in stream function formulation [91]

$$-\nu \Delta^2 \Psi + \Psi_y (\Delta \Psi)_x - \Psi_x (\Delta \Psi)_y = f \quad (27)$$

where Ψ denotes the stream function, ν the viscosity, and f the applied body force. A standard benchmark is the lid driven cavity problem [91, 92], which models flow in a unit square domain $\Omega = (0, 1)^2$, whose upper boundary moves to the right, whereas the rest of the boundaries are fixed. Corresponding boundary conditions are $\Psi = 0$ on all boundaries, $\Psi_x = 0$ and $\Psi_y = 0$ on the left, right and bottom boundaries (“no slip” requirement), $\Psi_x = 0$ and $\Psi_x = 1$ on the top boundary (the driven surface). At all boundaries, we impose Ψ strongly, while its derivatives are imposed weakly by a variant of Nitsche’s method [79, 81] with a penalty parameter $\beta = 50$. The nonlinearities in Equation (27) are handled by a fixed point iteration scheme [93].

The solution characteristics of the driven cavity problem are strongly determined by the choice of viscosity. We set $\nu = 1.25 \cdot 10^{-3}$, which results in a Reynolds number of $\text{Re} = 800$. A flow separation in the lower edges can thus be expected. Since we are also interested in accurately tracking the location of separation at $\Psi = 0$, we add an additional term to the gradient-based error indicator of Equation (24) as follows

$$\varepsilon_e = \frac{1}{V_e} \left[\left(\int_{\Omega_e} |\nabla \Psi|^2 \, d\Omega_e \right)^{1/2} + \left(\int_{\Omega_e} |\Psi|^2 \, d\Omega_e \right)^{-1/2} \right] \quad (28)$$

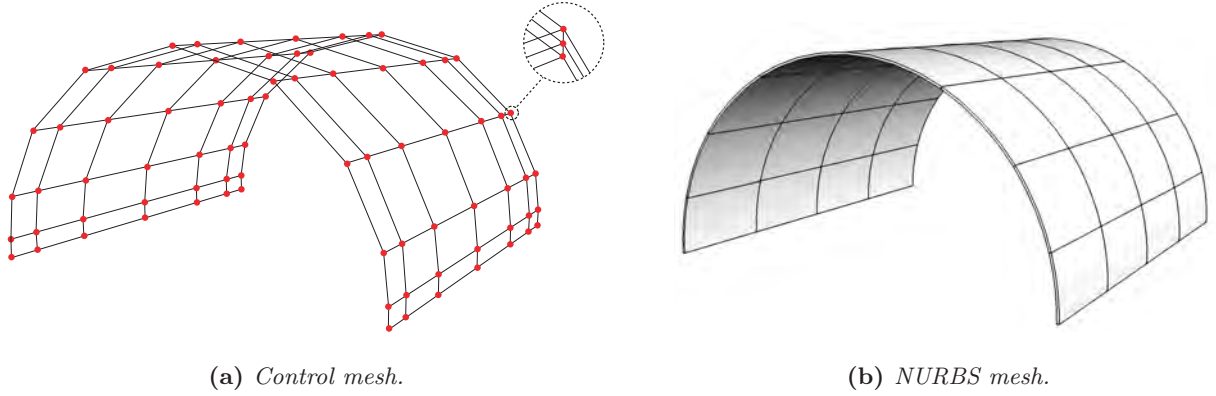


Figure 25: Geometry description of half a cylinder. The structure is modeled as a continuum with cubic NURBS in surface directions and 3 quadratic NURBS basis functions through the thickness.

so that ε_e also grows in the vicinity of $\Psi = 0$. The resulting hierarchical mesh of level $k = 4$, generated from an initial cubic B-spline discretization of 8×8 knot span elements, and the isolines of the corresponding stream function solution are plotted in Figure 24. They show that the refinement can be concentrated around the boundary layers and locations of flow separation. The adaptive mesh of Figure 24a with 3,517 dofs requires about 80% fewer degrees of freedom than a corresponding uniform discretization with 16,641 that uses the size of the smallest element of Figure 24a as the global mesh size.

6.3. NURBS geometries

According to the isogeometric paradigm, the NURBS basis of the geometry description should be also used for analysis, thus superseding geometry approximation and mesh generation of standard FEM [4, 5]. However, in most cases, the complexity in geometry and in the solution of the physical model do not coincide, so that the NURBS basis can represent the geometry exactly with a limited number of basis functions, while analysis requires local refinement to achieve satisfactory accuracy. The hierarchical refinement approach for NURBS based isogeometric analysis introduced in Section 5 maintains the initial geometry description by a given set of control points, weights and NURBS basis functions throughout the refinement process, while introducing additional levels of hierarchical basis functions, which achieve an adaptive refinement of the solution fields only.

6.3.1. The pinched cylinder

The first NURBS example is the pinched cylinder problem from the shell obstacle course [5, 94, 95]. Its geometry is given here by the control mesh and the corresponding mapped NURBS mesh of Figure 25, modeling one half of the cylinder. All geometrical mapping procedures throughout the adaptive analysis will revert to this model, which will not be refined, since it represents the geometry exactly. Following [5, 96], the shell is modeled as a three-dimensional solid and no shell assumptions are employed (see Figure 25a). The polynomial degree is cubic in the surface

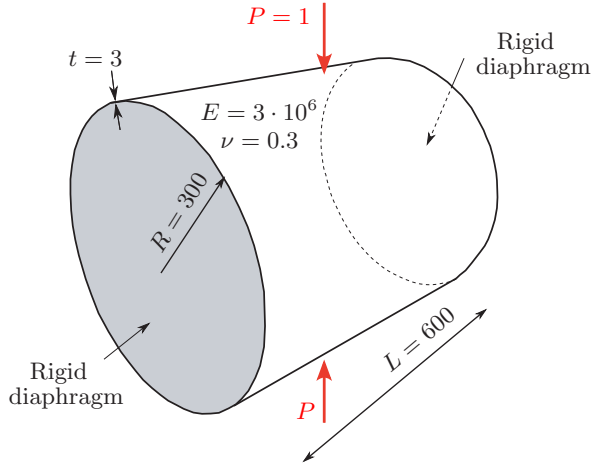


Figure 26: The pinched cylinder.

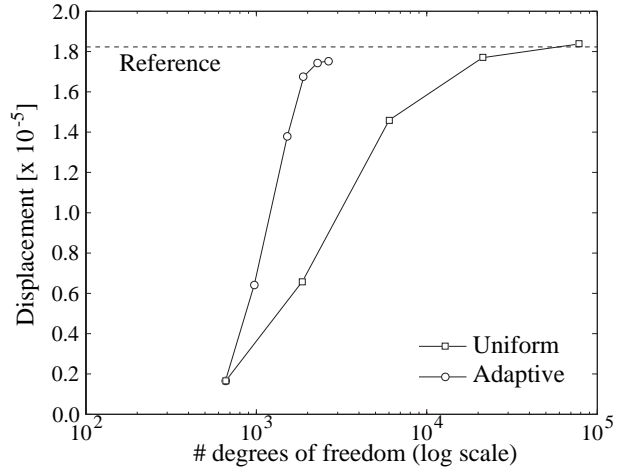
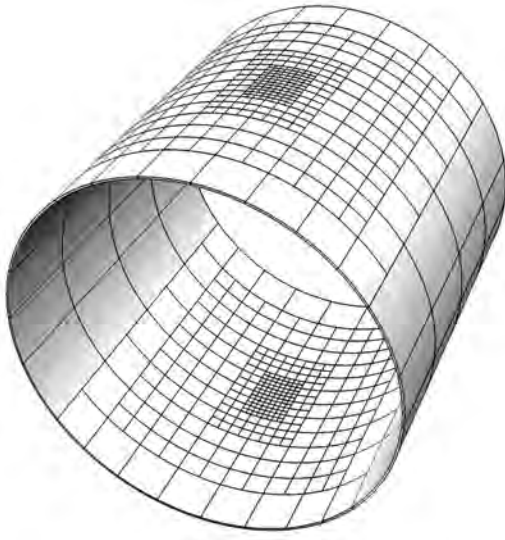
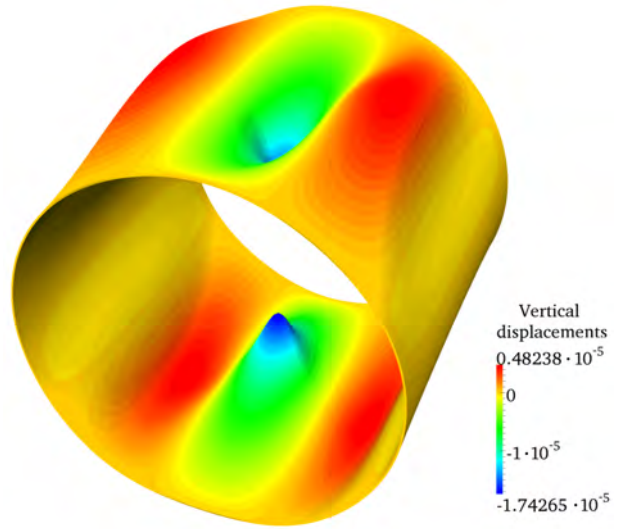


Figure 27: Convergence of the vertical displacement under the load.



(a) Hierarchical mesh of level $k = 4$.



(b) Corresponding vertical displacements plotted on the deformed structure.

Figure 28: Adaptive mesh and solution for the pinched cylinder problem. The results for the complete structure are obtained by mirroring the results obtained for the half cylinder.

directions, whereas only one quadratic NURBS element is used through the thickness, which has been shown to be adequate to obtain sufficiently accurate results [5, 97]. The problem definition of Figure 26 illustrates the external loading by two opposite concentrated forces, which results in highly localized deformation under the loads. At the longitudinal edges of the half cylinder, we apply symmetry boundary conditions. Automatic hierarchical refinement based on the concepts of Sections 4 and 5, with the error indicator of Equation (24), is applied to the initial discretization

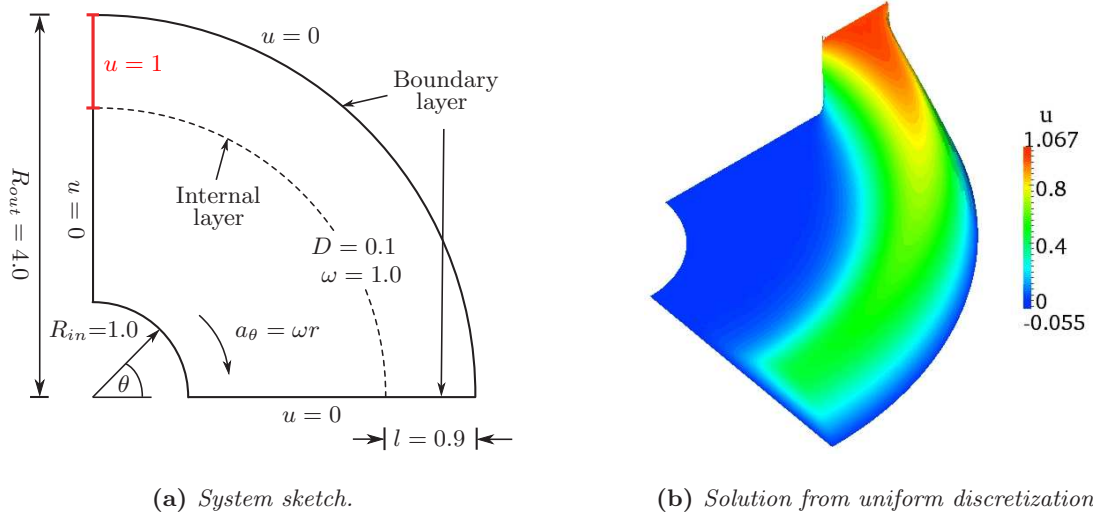


Figure 29: Advection-diffusion in an annular section.

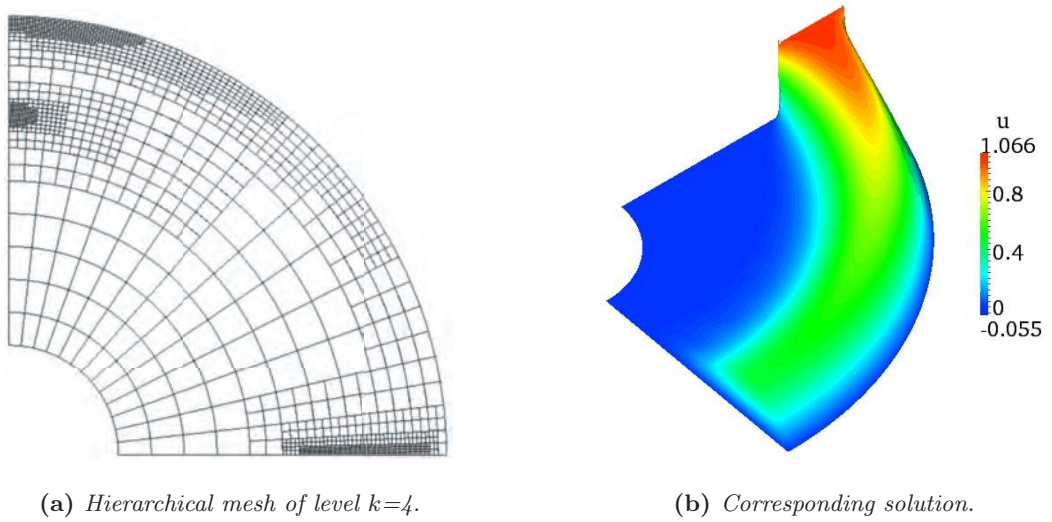


Figure 30: Adaptive mesh and solution for the 2D advection-diffusion problem.

of $8 \times 4 \times 1$ NURBS elements shown in Figure 25b. It carries out local refinement only in the surface directions and keeps one quadratic NURBS element in the thickness direction.

Convergence in terms of vertical displacements under the applied point load is shown in Figure 27, for which the analytical solution is given as $1.82488 \cdot 10^{-5}$ [5]. Uniform refinement converges to a slightly softer solution than the reference, which agrees well with previous results [5, 89]. In contrast, adaptive hierarchical refinement converges to a slightly smaller displacement. This can be attributed to the influence of the unrefined parts of the shell, which are not sufficiently resolved, but not detected by the simple gradient-based error indicator due to their relatively small gradients. Nonetheless, hierarchical refinement achieves a comparable level of accuracy with about one order

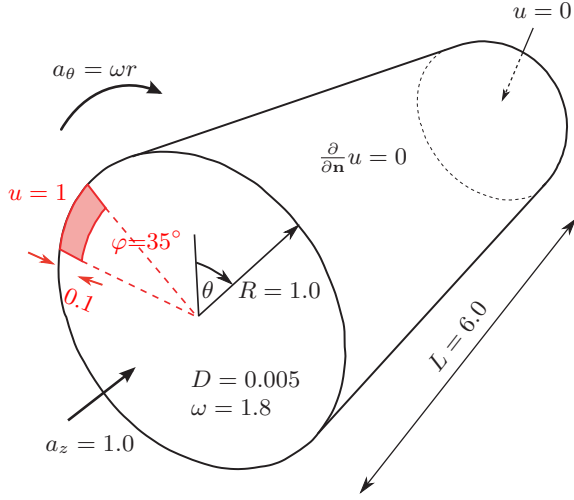


Figure 31: *Problem sketch.*

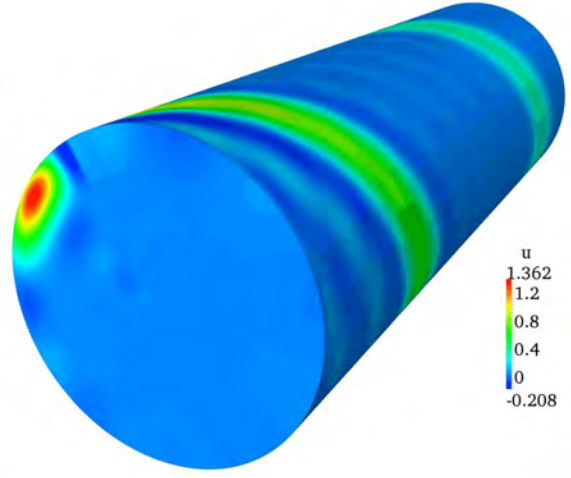
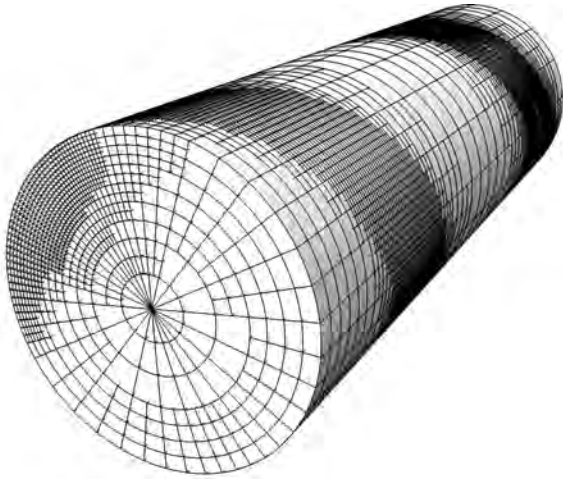
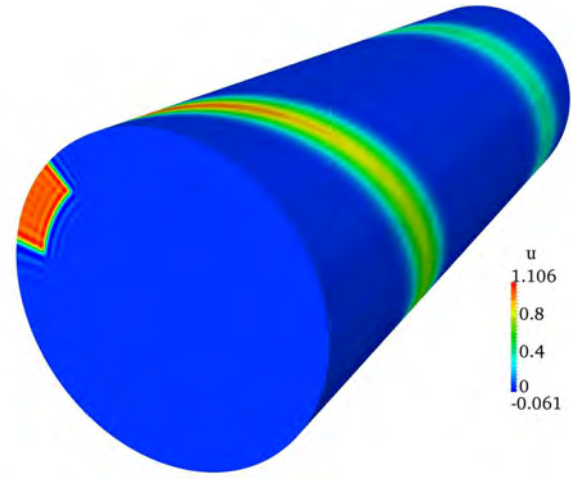


Figure 32: *Initial solution.*



(a) *Hierarchical NURBS mesh of level $k=3$.*



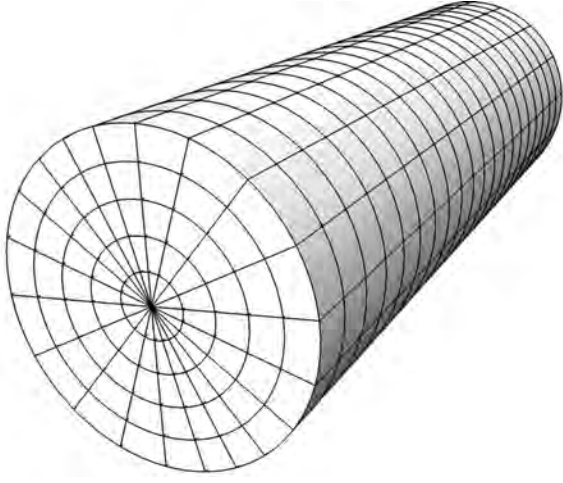
(b) *Corresponding solution.*

Figure 33: *Adaptive mesh and solution for the 3D advection-diffusion problem.*

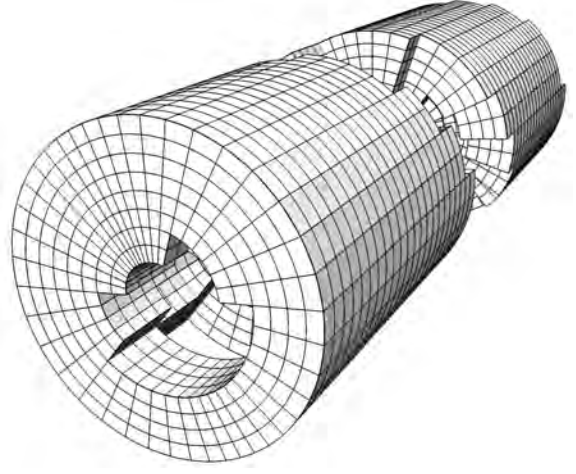
of magnitude fewer degrees of freedom than uniform refinement. The hierarchical mesh of level $k = 4$ and the corresponding displacement plot on the deformed structure are given in Figure 28, where deformations are largely magnified to clearly highlight of the deformation pattern.

6.3.2. Advection-diffusion in an annular section

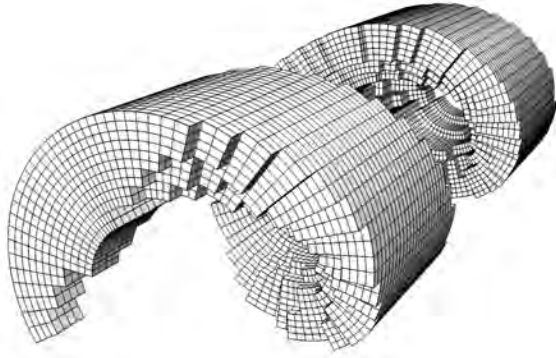
We focus again on linear advection-diffusion described by Equation (26). The present 2D example is defined over an annular quarter section, which is described exactly by an initial mesh of 10×13 cubic NURBS elements. The two-dimensional flow field corresponds to a rotational vortex with tangential velocity $a_\theta = \omega r$ and radial velocity $a_r = 0$. A sketch of the problem definition



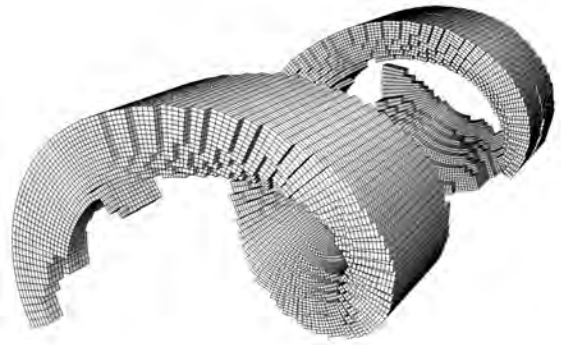
(a) Initial knot spans of $k=0$.



(b) Hierarchical knot spans of $k=1$.



(c) Hierarchical knot spans of $k=2$.



(d) Hierarchical knot spans of $k=3$.

Figure 34: Sequence of knot span elements, over which contracted basis functions of successive hierarchical levels k are defined. The combination of all levels results in the hierarchical NURBS mesh of Figure 33a.

is given in Figure 29a. Boundary conditions are prescribed weakly at the inflow and strongly at the outflow boundaries [79], where the penalty parameter β in Nitsche's method is 50. The Péclet number of this problem is $Pe = 10$. Over part of the inflow, the concentration u is set to 1, creating boundary and internal layers, which is illustrated by the reference solution of Figure 29b, obtained from a standard Galerkin discretization of 160×208 cubic NURBS elements. Applying automatic hierarchical refinement based on the gradient-based error indicator of Equation (24) and the initial standard Galerkin discretization, we obtain an adaptive mesh of level $k = 4$, plotted in Figure 30a. The corresponding solution in Figure 30b shows qualitatively no difference in terms of boundary and internal layers as well as absolute values in comparison to Figure 29b. However, the uniformly refined solution requires 33,810 degrees of freedom, whereas the adaptive mesh requires only 1,387,

but both provide the same resolution of the concentration jumps at the boundaries.

6.3.3. Advection-diffusion in a rotating cylinder

Finally, we consider advection-diffusion in a three-dimensional cylinder that rotates around its axis with tangential velocity $a_\theta = \omega r$ and radial velocity $a_r = 0$. At the same time, we assume a flow of constant axial velocity a_z , which results in a helical pattern of the concentration that emerges from the fixed local inflow boundary condition $u = 1$. The Péclet number of this problem is $Pe = 360$. The geometry of the cylinder is described exactly by two equal NURBS patches, each of which covers one half of the cylinder and consists of a standard Galerkin discretization of $5 \times 10 \times 20$ cubic NURBS elements in (r, θ, z) -directions, respectively (see Figure 34a). Boundary conditions are prescribed weakly at the inflow and strongly at the outflow boundaries [79] with a penalty parameter of $\beta = 50$. A sketch of the problem and the initial isogeometric solution are shown in Figures 31 and 32, respectively. The initial NURBS discretization is unable to accurately resolve the boundary and internal layers along the plume. Automatic hierarchical refinement based on the gradient-based error indicator of Equation (24) results in an adaptive mesh of level $k = 3$, plotted in Figure 33a. The corresponding solution in Figure 33b exhibits a considerably improved resolution of the boundary and internal layers. Figure 34 illustrates the sequence of knot spans, over which the contracted hierarchical basis functions are defined. The finest level $k = 3$ shown in Figure 34d demonstrates that the automatic refinement procedure is able to accurately track the revolving plume in the form of a helix. A uniform discretization that yields a plume resolution with the same element size requires a globally refined mesh of $40 \times 80 \times 160$ NURBS elements with 1,391,380 degrees of freedom, whereas the present adaptive mesh requires only 125,271 dofs.

7. The B-spline version of the finite cell method

Immersed boundary methods, also known as embedded domain methods, operate on a Cartesian fixed grid, which can be arbitrarily intersected by the physical boundary [57, 85, 98, 99]. They require special concepts for the accurate integration of elements cut by geometric boundaries, for the incorporation of Dirichlet boundary conditions [81, 100–103] and for the preservation of well-conditioned system matrices [35, 104, 105]. From a range of related approaches that combine the immersed boundary concept with axis-aligned grids of B-spline basis functions [34, 35, 80, 106, 107], we apply the recently introduced B-spline version of the finite cell method [60, 108] to illustrate the benefits of hierarchical refinement in the framework of B-spline based immersed boundary analysis. Before embarking on its hierarchical refinement in the next section, we briefly outline its principles and demonstrate its solution characteristics with a simple example in the following.

7.1. The fictitious domain approach and adaptive integration

The B-spline version of the finite cell method, which is a further development of the original p -version of the finite cell method (FCM) [58, 59, 61], combines the fictitious domain approach

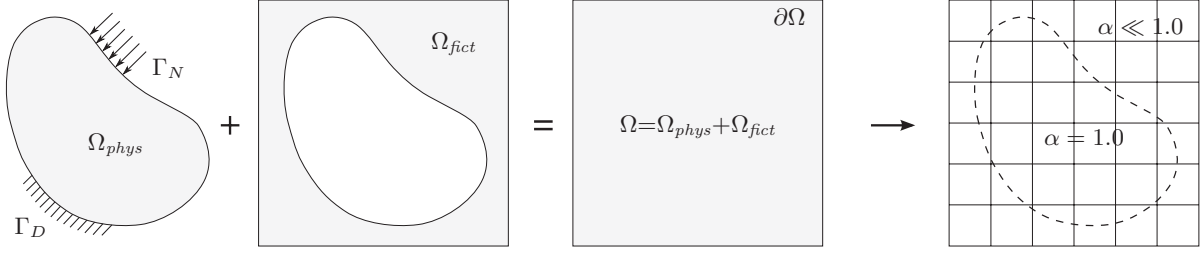


Figure 35: The fictitious domain concept: The physical domain Ω_{phys} is extended by the fictitious domain Ω_{fict} into the embedding domain Ω to allow easy meshing of complex geometries. The influence of Ω_{fict} is penalized by α .

with a regular Cartesian grid of axis aligned B-splines and an adaptive integration procedure for cut elements. Using the higher order and higher continuity of B-spline basis functions, it has been shown to preserve exponential rates of convergence under p -refinement for both linear and geometrically nonlinear structural problems, and to be well-suited for the analysis of voxel-based geometries of arbitrary complexity [60, 108].

Its main idea, illustrated in Figure 35, consists of the extension of the physical domain of interest Ω_{phys} beyond its physical boundaries into a larger embedding domain of simple geometry Ω , which can be meshed easily by a structured grid. To preserve consistency with the original problem, the influence of the fictitious domain extension $\Omega_{fict} := \Omega \setminus \Omega_{phys}$ is mitigated by penalizing its material parameters. In linear elasticity, this is achieved by complementing the elasticity tensor \mathbf{C} relating stresses and strains by a scalar factor α

$$\boldsymbol{\sigma} = \alpha \mathbf{C} : \boldsymbol{\varepsilon} \quad (29)$$

which leaves the material parameters unchanged in the physical domain, but penalizes the contribution of the fictitious domain

$$\alpha(\mathbf{x}) \begin{cases} = 1.0 & \forall \mathbf{x} \in \Omega_{phys} \\ \ll 1.0 & \forall \mathbf{x} \in \Omega_{fict} \end{cases} \quad (30)$$

Using a structured grid of knot span elements (see Figure 35), kinematic quantities are discretized with uniform B-splines, where all basis functions without support in the physical domain Ω_{phys} are omitted.

During numerical integration, elements cut by the geometric boundary are adaptively integrated by composed Gauss quadrature, based on a hierarchical decomposition of the original element into integration sub-cells [59, 60]. In d dimensions, the sub-cell structure can be built up in the sense of a d -dimensional tree [40]. Figure 36 outlines the sub-cell partitioning procedure for an example in 2D. Starting from the original element of level $k = 0$, each sub-cell of the current level $k = i$, $i > 0$, is first checked whether it is cut by a geometric boundary. This is simply achieved by a point location query, which determines for each integration point, whether it is located in Ω_{phys} or Ω_{fict} .

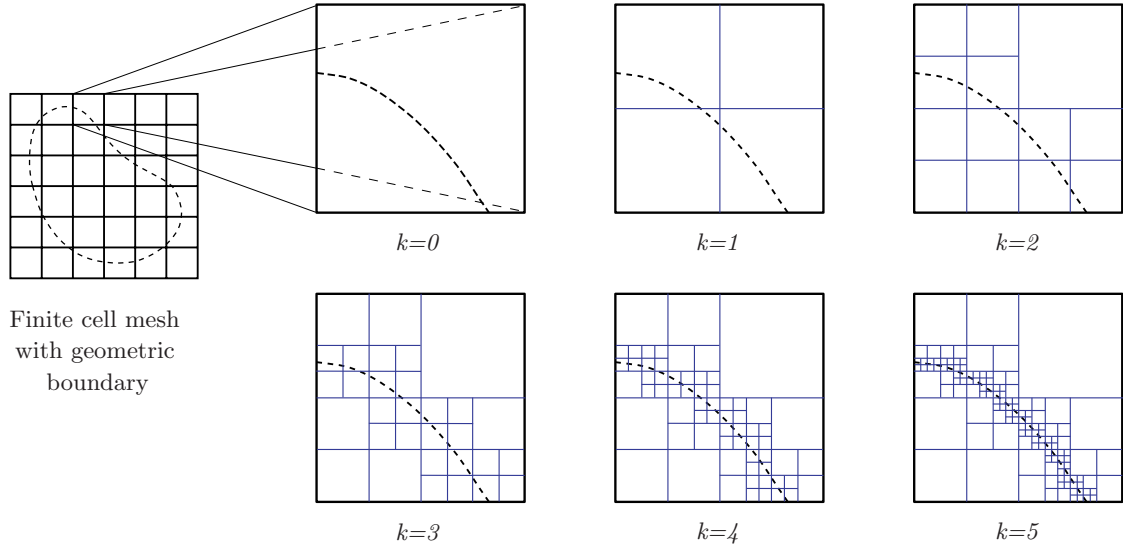


Figure 36: 2D sub-cell structure (blue lines) for adaptive integration of elements (black lines) cut by the geometric boundary (dashed line). The quadtree is built by iterating over the sub-cells of the current level k , until the max. depth is reached.

If integration points of the same element are located in both domains (hence a geometric boundary must be present), the element is replaced by 2^d equally spaced sub-cells of level $k = i + 1$, each of which is equipped with $(p+1)^d$ Gauss points. Partitioning is repeated for all sub-cells of the current level k , until a predefined maximum depth $k = m$ is reached. This approach keeps its simplicity in the presence of arbitrarily complex boundaries, while its implementation is straightforward. While elements completely outside the physical domain can be neglected, integration points of all sub-cells must be taken into account with a finite value of α to prevent extreme ill-conditioning of the stiffness matrix [58, 59]. Depending on the accuracy required, values of α may range between 10^{-3} and 10^{-15} .

7.2. Solution characteristics of the B-spline version of the FCM

The FCM solution behavior is briefly described with the help of the example of an infinite plate with a circular hole under in-plane tension. Geometry, material, boundary conditions and the analytical solution [109] correspond to the classical example shown in [1, 5] and are given in Figure 37. Due to symmetry, only one quarter of the problem is considered.

Figure 38 illustrates the discretization procedure in the framework of the immersed boundary concept. B-spline basis functions are defined over an axis-aligned knot span mesh, which embeds the physical domain Ω_{phys} of the plate in a simple square. The circular hole constitutes a fictitious domain Ω_{fict} , whose influence is mitigated by penalizing the elasticity matrix at integration point level with a finite value $\alpha=10^{-12}$ according to Equations (29) and (30). The integration accuracy in cut elements is ensured by adaptive integration sub-cells, which lead to an aggregation of Gauss points around the boundary of the circular hole. It is important to note that integration sub-cells

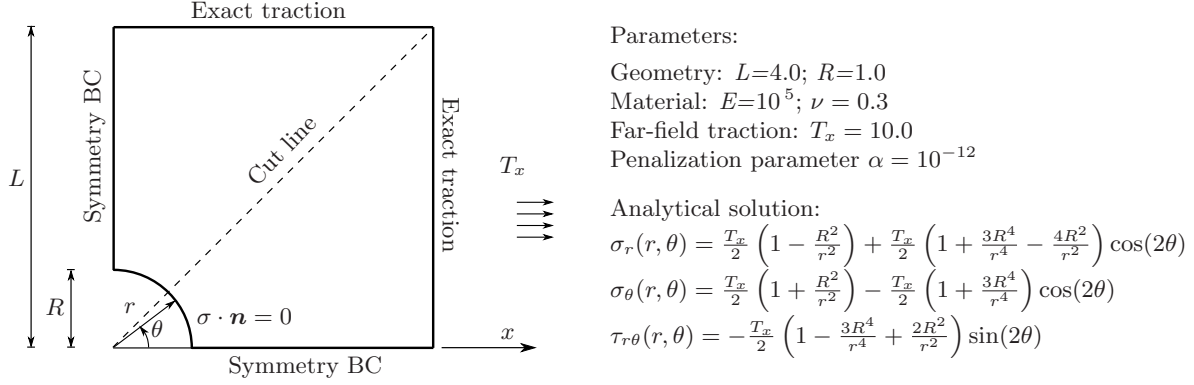


Figure 37: Linear elastic plate with a circular hole: problem definition and exact solution.

do not affect the B-spline basis functions, but only increase the number of integration points around the geometric boundary. The element located completely outside Ω_{phys} is not integrated. Basis functions with support in the lower left element only are eliminated from the basis, while Dirichlet constraints can be incorporated strongly here.

The accuracy of the B-spline version of the FCM relies on the smooth extension of the solution fields into the fictitious domain, illustrated in Figure 39 by von Mises equivalent strains plotted along the diagonal cut line (see Figure 37). While the exact solution is only defined over the physical domain Ω_{phys} , the immersed boundary solution extends smoothly into the fictitious domain Ω_{fict} , although the elasticity matrix is discontinuous due to the penalization factor α . A tentative explanation for this important characteristic can be found by considering the total strain energy

$$U = \int_{\Omega} \Psi dV = \frac{1}{2} \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\varepsilon} dV \quad (31)$$

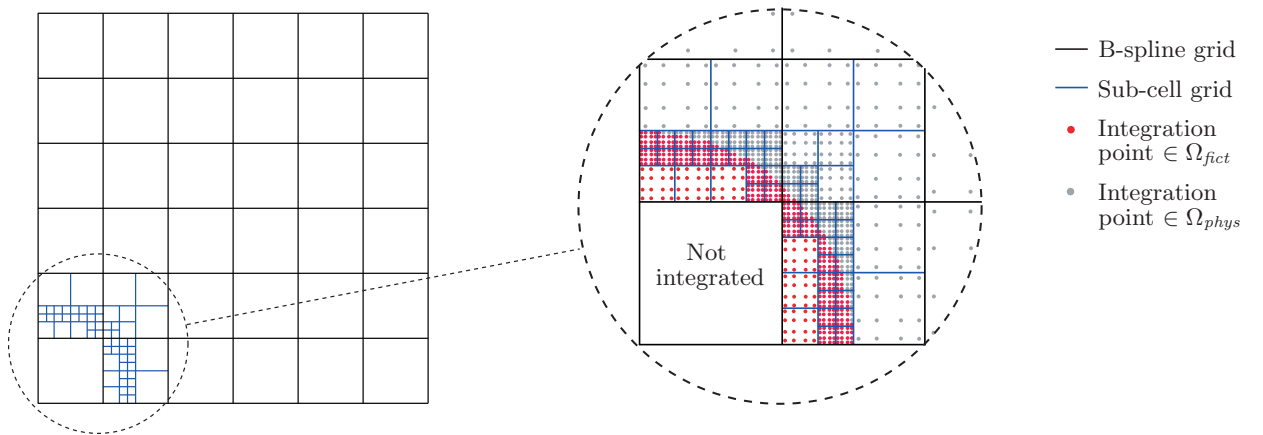


Figure 38: B-splines are defined over an axis-aligned grid of knot spans (black lines). A sub-cell grid (blue lines) leads to an aggregation of Gauss points around the boundary, where the elasticity matrix is penalized according to Equation (29) at all points outside the physical domain Ω_{phys} (marked in red).

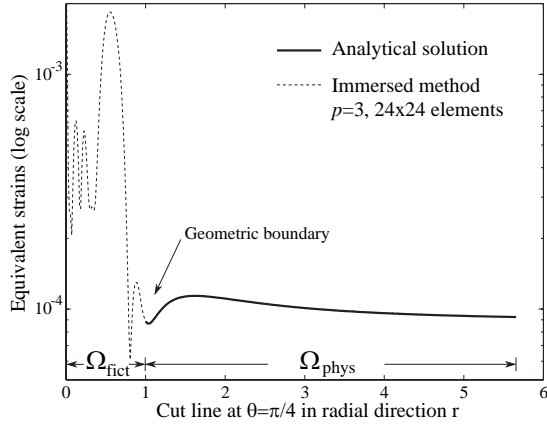


Figure 39: Von Mises equivalent strains plotted along the diagonal cutline (see Figure 37).

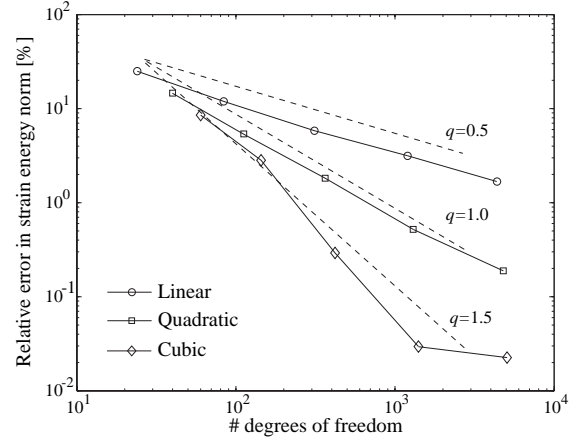


Figure 40: Convergence in energy norm, when the B-spline grid is uniformly refined.

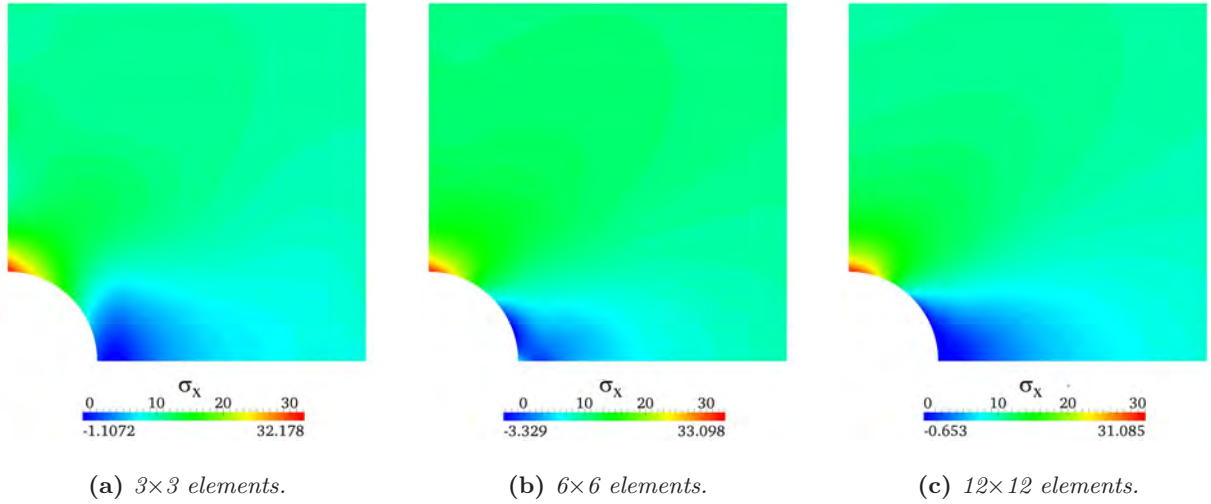
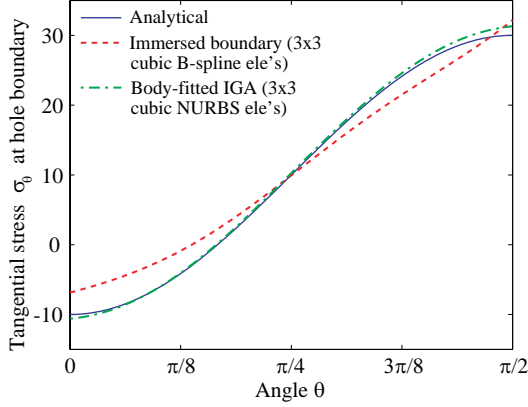


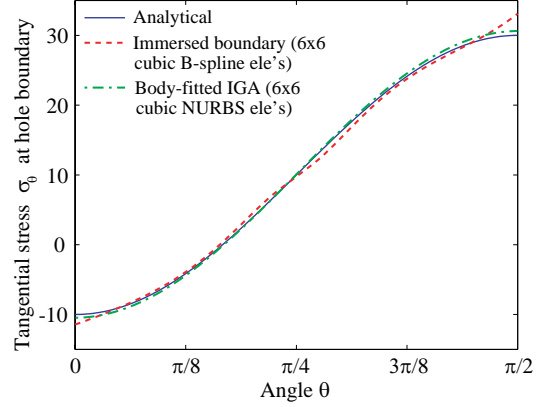
Figure 41: Normal stresses σ_x , plotted over Ω_{phys} , for different cubic B-spline meshes.

where Ψ represents the strain energy function, defined over the complete domain Ω . The best approximation property to the total strain energy U states that the solution of a Galerkin finite element scheme represents a least-squares best fit to the exact solution [110, 111]. Due to the penalization with α , which is present in σ of Equation (31) due to Equation (29), deviations from the exact solution in the fictitious domain Ω_{fict} have a considerably smaller impact on the strain energy than deviations in the physical domain Ω_{phys} . Therefore, a minimization of the strain energy error by the B-spline basis of the immersed boundary scheme results in an accurate approximation in the physical domain Ω_{phys} .

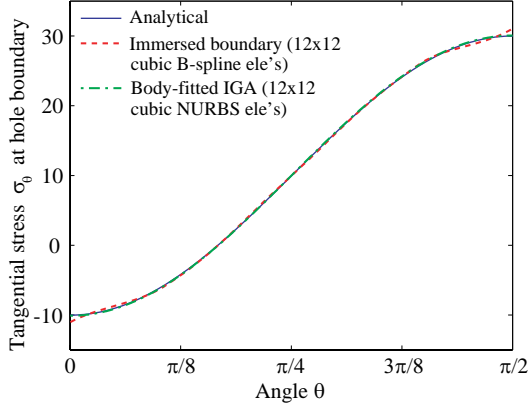
In particular, this implies a smooth extension of the solution fields into the fictitious domain,



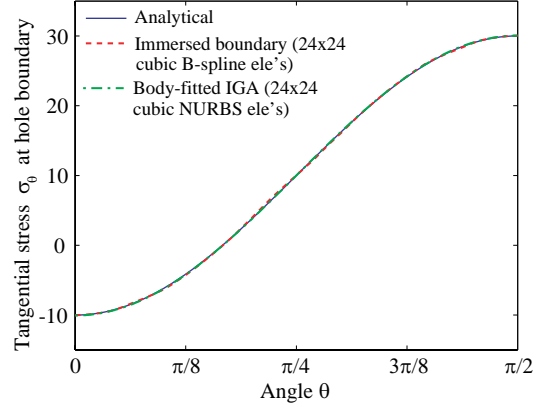
(a) 3×3 elements.



(b) 6×6 elements.



(c) 12×12 elements.



(d) 24×24 elements.

Figure 42: Convergence of circumferential stresses $\sigma_\theta(R, \theta)$ at the circular boundary of the hole ($R = 1.0$) for the immersed boundary case and the body-fitted IGA case [5].

so that its gradients in the physical domain remain accurate up to the geometric boundary (see Figure 39). Furthermore, the solution can converge uniformly all over the physical domain, so that optimal rates of convergence can be achieved. Figure 40 shows the convergence behavior in strain energy under uniform h -refinement [110, 111], starting from a mesh of 3×3 B-spline elements. Note that the strain energy of the immersed method is computed by taking into account contributions of Ω_{phys} only. For linear, quadratic and cubic B-splines, the optimal rates of 0.5, 1.0 and 1.5 are met. Another benefit of the smooth extension property is the accuracy in stresses that can be achieved with immersed boundary methods. Figure 41 shows plots of the normal stress in horizontal x -direction for different cubic immersed boundary meshes, which can be compared to corresponding plots given in [5] that were obtained with body-fitted NURBS discretizations. While the two coarsest meshes of 3×3 and 6×6 elements distinctly differ from the expected solution, the

stress pattern of the 12×12 mesh shows no difference to the best solution shown in [5]. Of particular interest in this respect is the stress accuracy that can be achieved directly on the geometric boundary cutting through elements. We therefore plot the circumferential stress component on the circular boundary of the hole obtained with different cubic immersed boundary meshes in Figure 42a through 42d. We compare the immersed boundary results to stresses that were obtained with corresponding body-fitted cubic NURBS discretizations generated in the same as shown in [5]. The plots illustrate that the immersed boundary stresses converge to the analytical reference. A comparison with the standard IGA results reveals that the stresses of the body-fitted NURBS discretization are more accurate on coarse meshes as might be expected.

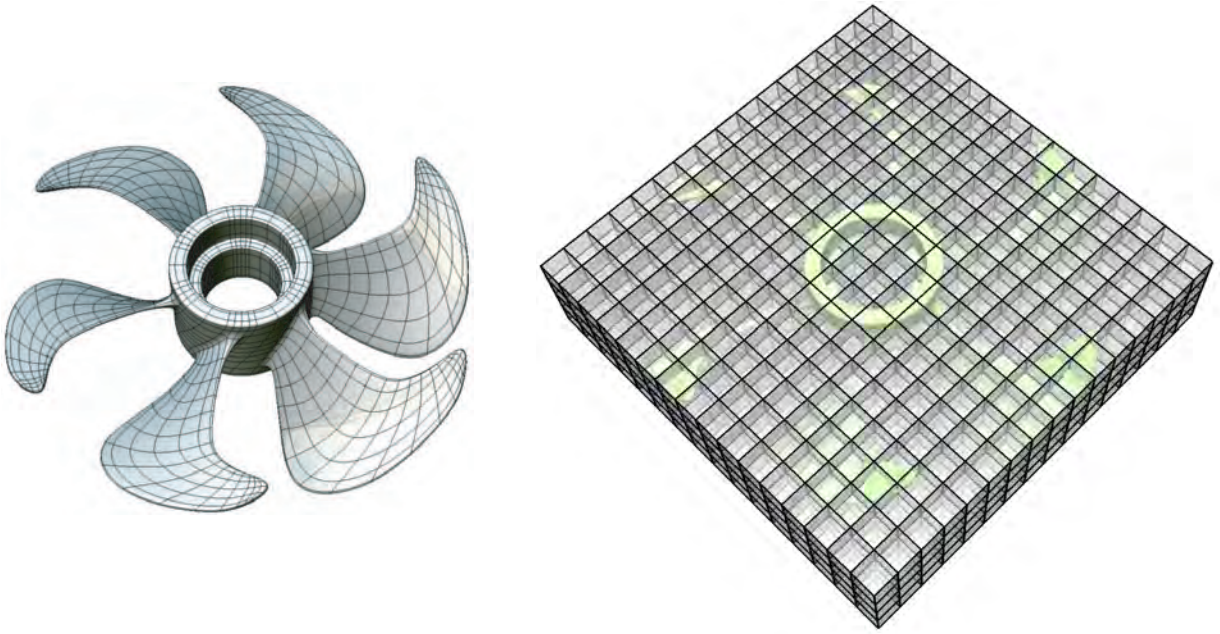
8. Hierarchical refinement and immersed boundary methods

The translation of complex CAD based geometrical models into conforming finite element discretizations is computationally expensive, difficult to fully automate and often leads to error-prone meshes, which have to be improved manually by the user. Immersed boundary methods do not require body-fitted meshes, but embed the domain into a regular grid of axis-aligned elements, which can be generated irrespective of the geometric complexity involved. The corresponding meshing procedure, which is based on a simple point location query, requires no user interaction and thus can be fully automated. This opens the door to a seamless IGA design-through-analysis procedure for complex engineering parts and assemblies, which we demonstrate in the following by the examples of a ship propeller and a rim of an automobile wheel. We apply the B-spline version of the finite cell method, reviewed in Section 7, for modal and stress-displacement analysis. We furthermore demonstrate that hierarchical refinement of B-splines can considerably increase the flexibility of the method by adaptively resolving local features in geometry and solution fields while thanks to its straightforward implementation, the key benefit of full automation can be maintained.

8.1. Modal analysis of a ship propeller

The geometry of the propeller is given by a smooth, watertight T-spline surface (i.e., there are no gaps or overlaps). It is exported from the CAD package Rhino [66] in conjunction with the T-spline plug-in [112] in the form of Bézier elements as shown in Figure 43a. Its maximum diameter and height is 0.695 m and 0.334 m, respectively, and it is made out of steel with Young's modulus $2.1 \cdot 10^{11}$ N/m², Poisson's ratio 0.28 and density 7,850 kg/m³. The thickness of the cylindrical hub in the center is about four times larger than the average thickness of the surrounding propeller blades. The structure can neither be characterized as a typical shell nor as a true solid. Configurations like this usually require specialized and time consuming meshing procedures to produce good quality discretizations.

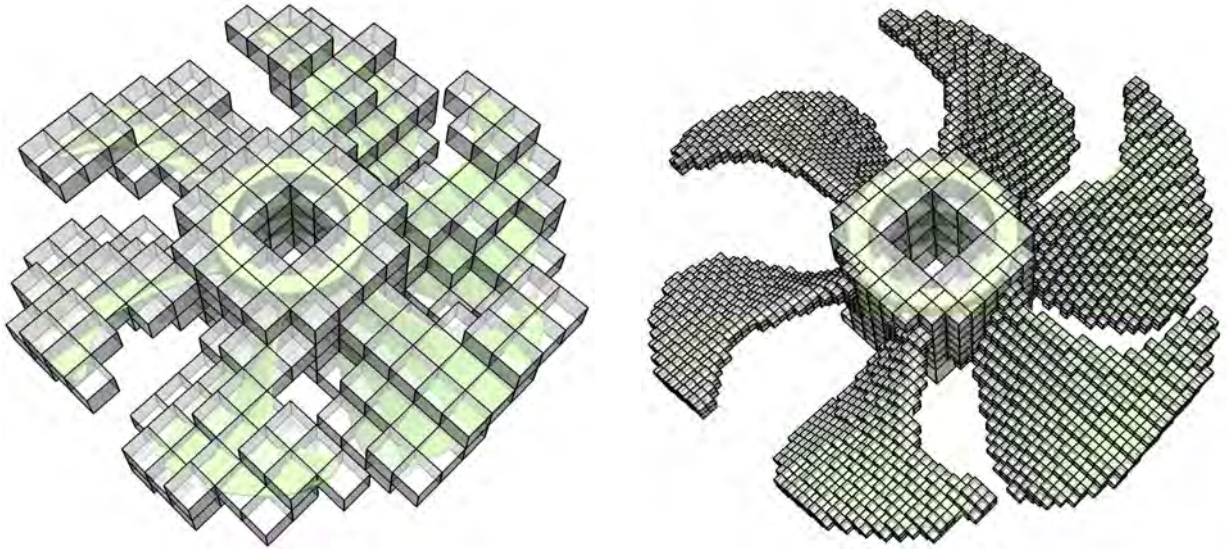
We apply the B-spline version of the finite cell method to illustrate the discretization procedure in the framework of the immersed boundary concept and its combination with hierarchical



(a) Bézier elements of a T-spline surface (Output from CAD package Rhino with T-spline plug-in).

(b) The complete structure is immersed in a bounding box of $16 \times 16 \times 4$ axis-aligned cubic B-spline elements.

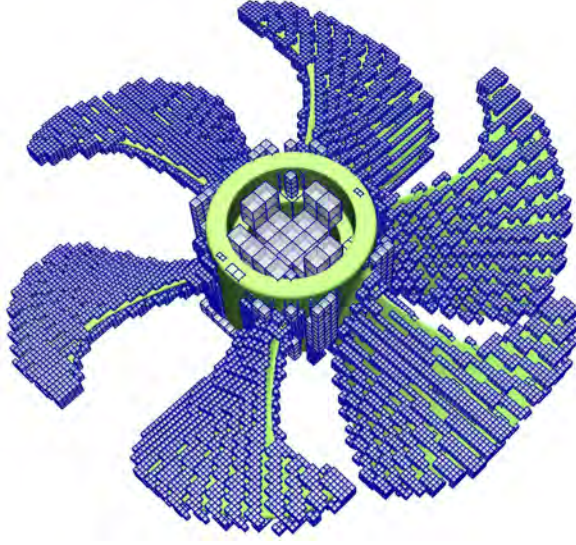
Figure 43: Propeller example: CAD based geometry description and immersed boundary discretization.



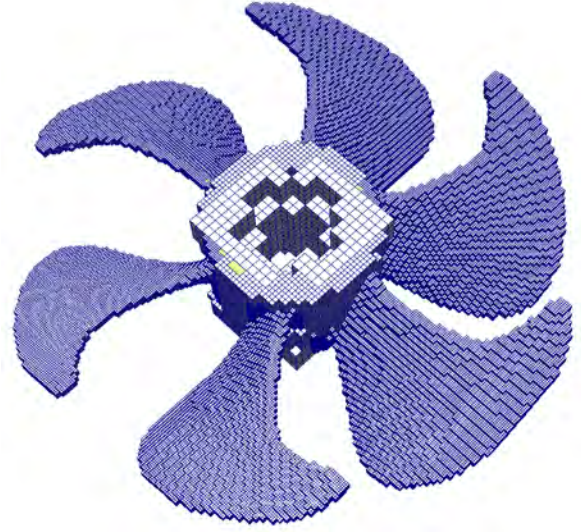
(a) Deletion of elements without support in the propeller domain creates a reduced set of elements, which homogeneously resolve the structure irrespective of the local thickness.

(b) Hierarchical refinement of the propeller blades achieves a homogeneous through-the-thickness resolution. Corresponding elements are octasected twice.

Figure 44: Propeller example: The role of hierarchical refinement



(a) 18,499 sub-cells of level $k=1$.



(b) 226,272 sub-cells of level $k=2$.

Figure 45: Propeller example: Sub-cell partitioning of elements cut by the geometric boundary. The partitioning scheme shown in Figure 38 is carried out up to level $k=2$.

refinement. First, the complete structure is embedded in a regular grid of axis-aligned B-splines as illustrated in Figure 43b. In this example, we apply uniform B-splines of polynomial degree $p=3$, which offer higher order approximation, but still are computationally efficient due to their relatively local support. Second, all those knot span elements without support in the propeller domain Ω_{phys} are eliminated from the discretization, which leads to a reduced set of elements displayed in Figure 44a. The decision whether an element is to be kept or not is based on a simple point location query, which checks if at least one integration point is located in Ω_{phys} . An efficient point location query can be achieved for example by a voxelization of the embedding domain [113, 114], where a boolean flag indicates for each voxel whether it is located in- or outside of the physical domain, or by search algorithms based on special space-partitioning data structures such as k -d trees [40, 115, 116].

An axis-aligned discretization with elements of the same size does not account for the inhomogeneous thickness of the different regions of the structure. In a third step, we therefore apply two levels of hierarchical refinement to the propeller blades, while we leave the discretization of the central hub as it is, in order to achieve a homogeneous resolution of the two different thicknesses. Whereas a uniformly refined mesh of the propeller that provides the same resolution of the blades has 80,922 dofs, the adaptive mesh shown in Figure 44b has only 53,052 dofs. In a fourth step, we equip each element cut by the geometric boundary by additional sub-cells, which are organized in an octree of depth two, generated by the partitioning procedure described in Section 7.1. The sub-cells corresponding to the leaves of the first and second levels of the octree are shown in Figure 45a

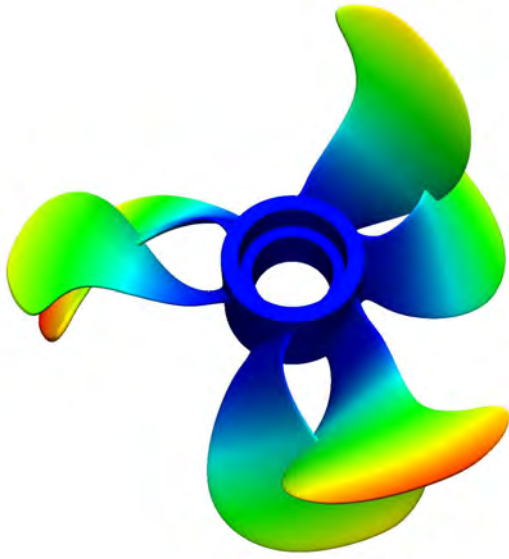
and 45b, respectively. It should be noted again that the blue sub-cells of Figure 45 do not affect B-spline basis functions, which are still defined over the set of elements shown in black lines in Figure 44b. Each sub-cell contains $4 \times 4 \times 4$ Gauss points, leading to an aggregation of integration points in cut elements to accurately take into account the geometric boundary during numerical integration. If a point location query indicates that an integration point is located outside the propeller domain Ω_{phys} , its contributions to the stiffness matrix and the mass matrix are penalized by factor $\alpha=10^{-3}$ in the sense of Equations (29) and (30).

The hierarchically refined mesh of Figure 44b is analysis suitable and is used in combination with the sub-cells of Figure 45 to conduct a modal analysis of the structure, where the mass matrix is lumped according to the row sum method [110]. For the solution of the Eigenvalue problem, we use the library package ARPACK [117]. Figure 46 illustrates the first seven mode shapes. The three lowest mode shapes, each of which corresponds to one pair of opposing blades, exhibit a rotational symmetry in the propeller plane around the center of the hub. The next two mode shapes are the same, but the second is rotated 90° compared to the first within the plane of the propeller. They are thus symmetric with respect to the two axes that span the propeller plane. The sixth mode shape is single and denotes a rotationally symmetric bending of the blades out of the propeller plane. The seventh mode shape shows an out-of-plane bending of two opposite blades. The reflection of geometric symmetries in the mode shapes corresponds very well to engineering experience and indicates a good quality of the first modes.

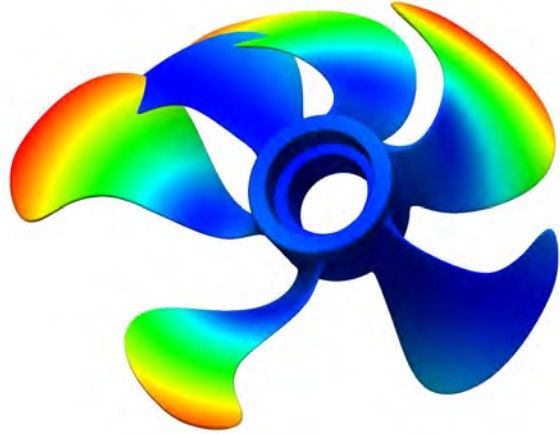
8.2. Stress-displacement analysis of the rim of an automobile wheel

The second example is a rim of an automobile wheel. Analogous to the propeller, its geometry is originally described by a T-spline surface, which is exported from the CAD package Rhino [66] using the T-spline plug-in [112] in the form of Bézier elements as shown in Figure 47a. Its maximum diameter and height are 239.26 mm and 223.28 mm, respectively, and its material is aluminium with Young's modulus $7.0 \cdot 10^4$ N/mm² and Poisson's ratio 0.34. We apply an ellipsoidal line load at the bottom outer edge of the rim in the direction of the global y -axis, and homogeneous Dirichlet constraints at the hub fix the structure, simulating the suspension of the wheel. Horizontal loadings typically occur due to centrifugal forces, when the vehicle enters into a curve, and are transferred from the road surface onto the outer edge of the rim by the tire. Boundary conditions are further specified in Figure 47b.

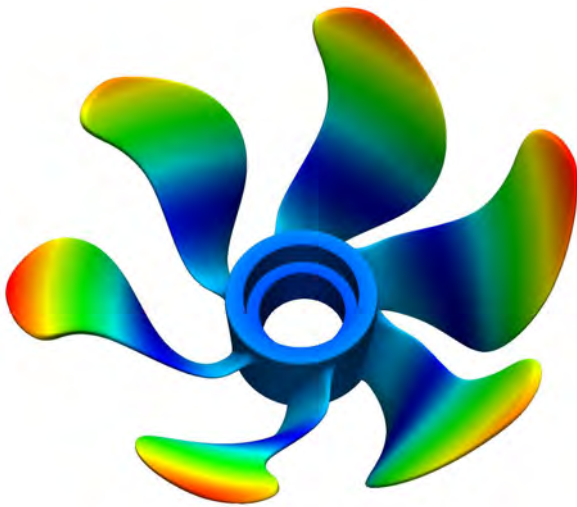
The discretization procedure in the framework of the finite cell method is illustrated in Figure 48. In a first step, the structure is completely immersed into a cuboidal bounding box of axis-aligned knot span elements, over which uniform B-splines are defined (Figure 48a). In a second step, elements located completely outside the physical domain Ω_{phys} of the wheel are erased from the discretization (Figure 48b). The location of the elements is determined by the straightforward point location query based procedure described in Section 8.1. In a third step, we apply hierarchical refinement to adaptively resolve those parts of the structure, where we expect larger



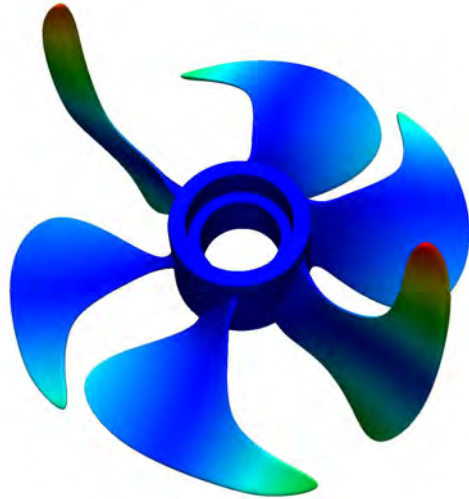
(a) The first, second and third mode shapes exhibit a rotational symmetry around the center, corresponding to the three pairs of opposing propeller blades. We display mode 2.



(b) The fourth and fifth mode shapes exhibit a symmetry with respect to the two axis of the propeller plane. We display mode 5.

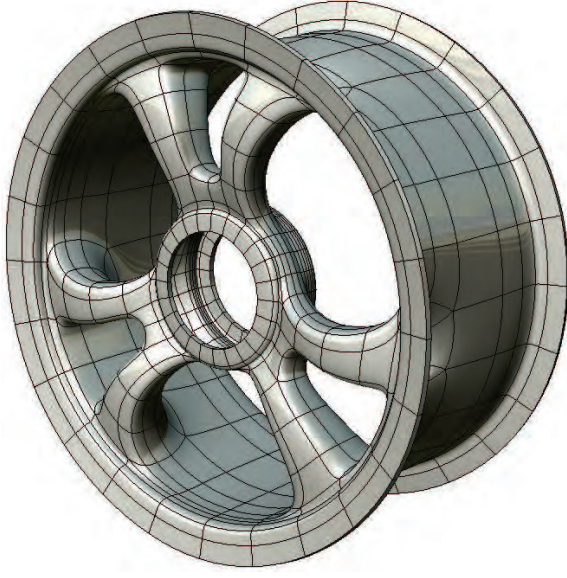


(c) The sixth mode shape is single and shows bending of the blades out of the propeller plane.

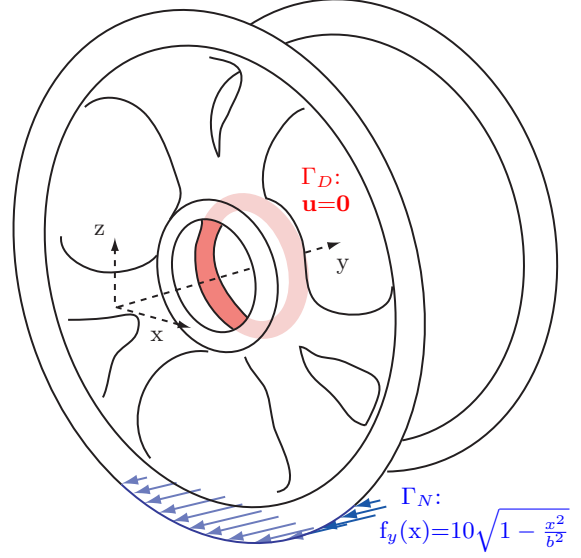


(d) The seventh mode shape shows the bending of opposing blades out of the propeller plane.

Figure 46: Modal analysis of the ship propeller. Most of the activity occurs in the propeller blades, whose resolution has been adaptively increased by hierarchical refinement. The color scale refers to the absolute displacement (dark blue - no displacement, red - highest displacement).



(a) Bézier elements of a T-spline surface (Output from CAD package Rhino with T-spline plug-in).

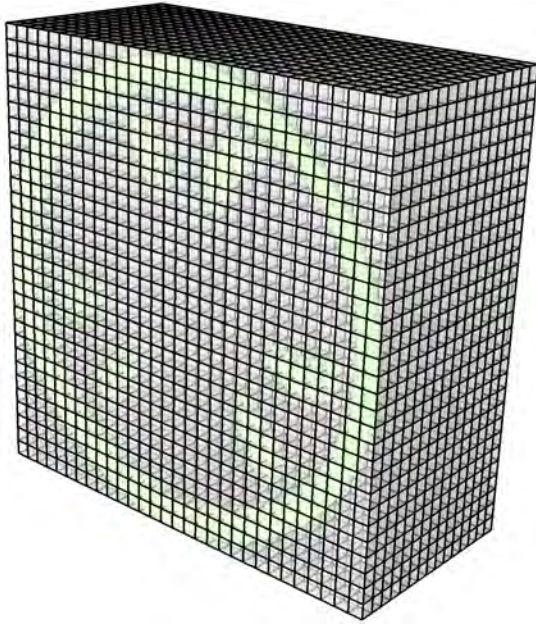


(b) Horizontal loading on the outer edge of the rim, fixture at the hub of the wheel. (Radius $R=239$ mm and $b=R \sin \pi/5$)

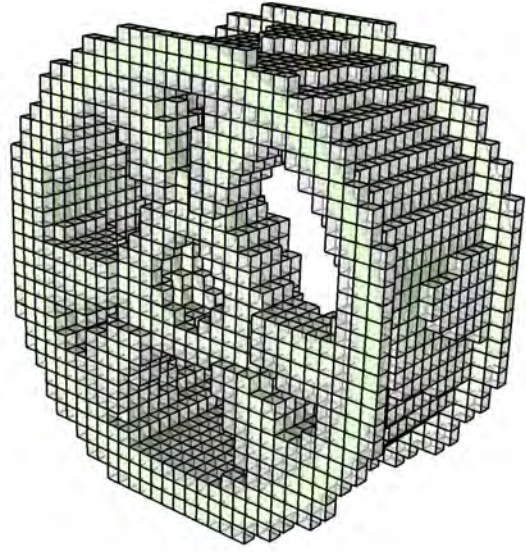
Figure 47: Rim of an automobile wheel: Geometry description and boundary conditions.

gradients in the solution fields (Figure 48c). The first level of refinement addresses the complex geometric parts of the spokes, the central hub and the upper part of the tire bearing. The second level of refinement is applied to the lower part of the tire bearing, where the loading is imposed. In a last step, we apply the simple partitioning procedure of Section 7.1 to elements cut by the geometric boundary, which generates two levels of sub-cells that accurately resolve the geometric boundary by integration points (Figure 48d). The stiffness contribution of integration points located outside the rim domain are penalized by factor $\alpha=10^{-6}$ in the sense of Equations (29) and (30).

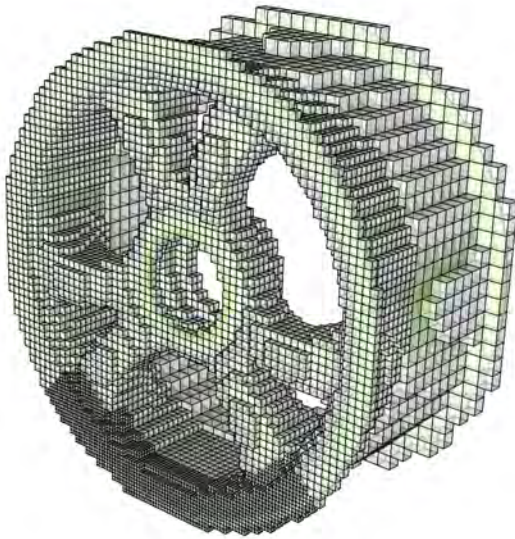
Dirichlet constraints are considered in a weak sense by applying Nitsche's method, whose variational formulation requires an integration over the Dirichlet boundary. Taking advantage of the axis-aligned Dirichlet surface Γ_D shown in Figure 47b, corresponding integrals can be evaluated by a projection of Gauss points onto Γ_D in each sub-cell that cuts Γ_D . The discrete system of equations is passed to the direct solver Pardiso [118]. The resulting deformation of the rim is illustrated in Figure 49, where the displacements imposed on the structure are magnified by a factor of 300 for better visibility. It can be observed that the solution in the vicinity of the loading area at the outer edge of the rim exhibits steep gradients, which confirm the necessity of local hierarchical refinement there. The adaptive mesh of Figure 48c with 105,807 dofs considerably decreases the computational effort in comparison to uniform refinement, which requires 622,989 dofs for the same resolution of the loading area. Figure 50 shows von Mises stresses, plotted on the boundaries of



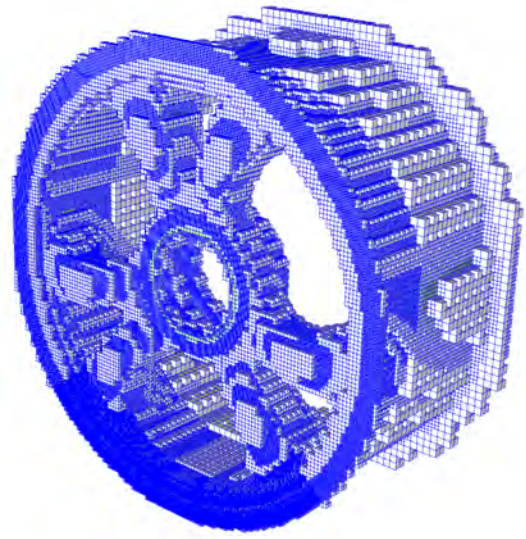
(a) The complete structure is immersed in a bounding box of $30 \times 15 \times 30$ axis-aligned B-spline elements.



(b) Elements without support in the physical domain of the rim are deleted.



(c) Hierarchical refinement adaptively increases the resolution, where high gradients of the solution are expected.



(d) Cut elements are adaptively partitioned twice for accurate integration of the geometric boundary (473,641 sub-cells).

Figure 48: Immersed boundary discretization of the rim of an automobile wheel with the B-spline version of the finite cell method and hierarchical refinement for adaptive stress-displacement analysis.

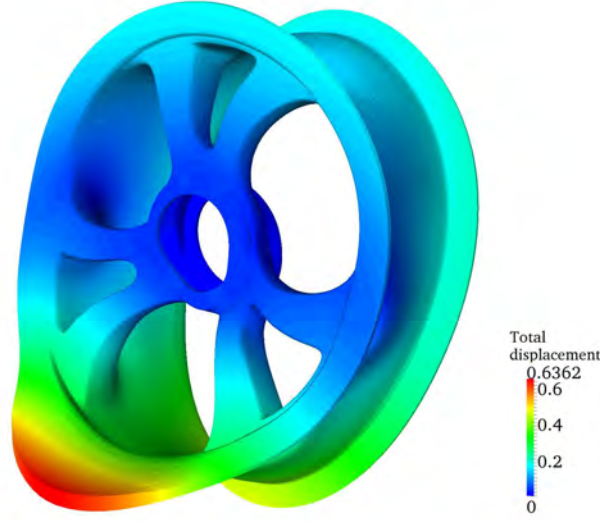


Figure 49: Total displacements in mm, plotted on the deformed structure. Steep gradients of the solution around the loading area on the lower edge have been adaptively resolved by hierarchical refinement.

the undeformed structure. Large stresses occur predominantly around the lower tire bearing, the two lower spokes and the central hub, thus tracing the flow of forces from the loading area to the support, while the rest of the structure remains unstressed. According to the symmetry in geometry and boundary constraints, the stress solution is completely symmetric and exhibits the typical stress concentration phenomena at reentrant sharp curves (see front view of Figure 50a). Due to the superposition of tensile membrane and bending stress, the maximum von Mises stress occurs at the sharp reentrant bend, where the loaded boundary ring bends into the main cylinder of the tire bearing (bottom-up view of Figure 50b).

9. Summary and conclusions

In this paper, we derived a hierarchical refinement procedure based on the concept of B-spline subdivision, which combines full analysis suitability with direct generalization to higher dimensions, in particular 3D, and a straightforward implementation in tree data structures. We discussed in detail theoretical concepts as well as algorithmic and implementation aspects for hierarchical refinement of B-splines, and presented their extension to NURBS. Using some elementary fluid and structural analysis problems, we successfully tested hierarchical refinement as a basis for adaptive NURBS based isogeometric analysis. We found that the method leads to adaptive meshes with highly localized refinement, and we experienced no mesh propagation away from the areas of interest. Comparing adaptive hierarchical with standard uniform refinement, we observed that the computational effort in terms of degrees of freedom is in general reduced about one order of magnitude at a comparable level of accuracy. We demonstrated that these beneficial characteristics carry over fully to three-dimensional solid NURBS elements. We therefore believe that hierarchical

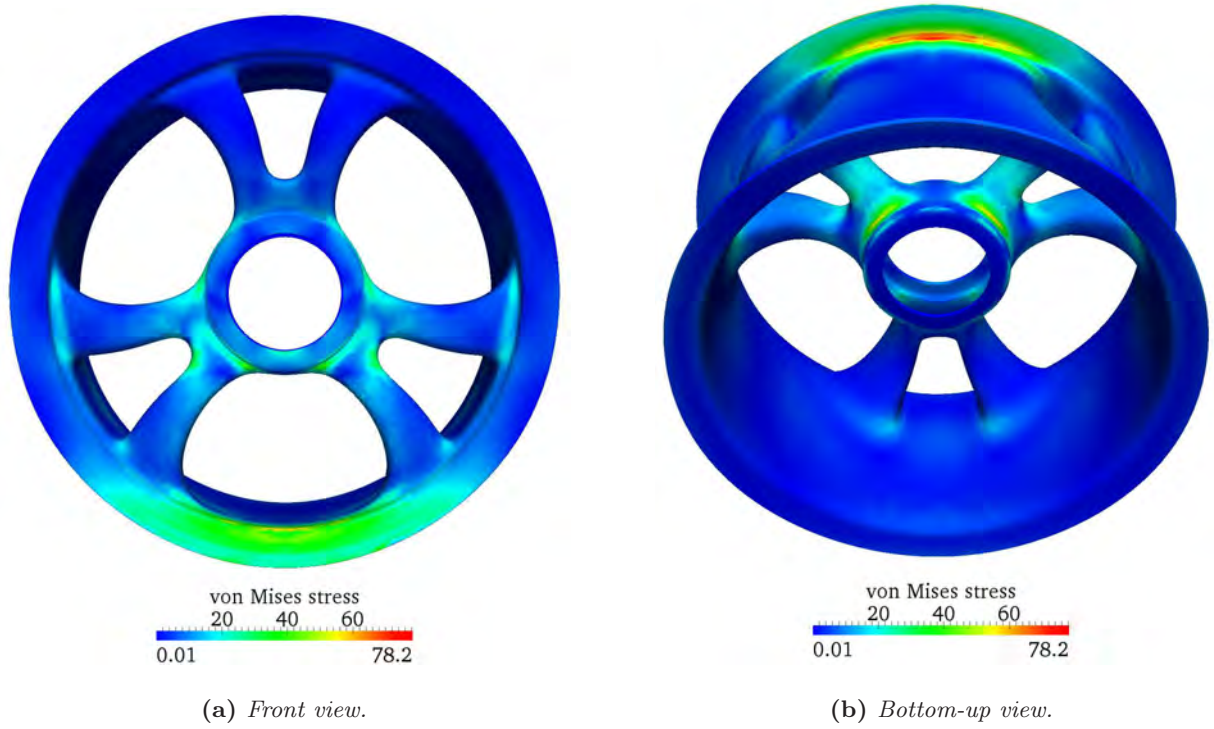


Figure 50: *Von Mises stress in the rim of an automobile wheel.*

refinement in this form has great potential to establish itself as an efficient and versatile technique for local refinement in NURBS based isogeometric analysis.

We also explored the application of hierarchical refinement of B-splines in the framework of immersed boundary analysis. From a range of related methods that combine B-spline approximations with immersed boundary methods, we chose the B-spline version of the finite cell method, for which we provided a concise review. In particular, we illustrated that this approach achieves optimal rates of convergence and is able to yield accurate stress results not only within the domain of interest, but also directly on the immersed boundary. We then applied the B-spline version of the finite cell method to a ship propeller and a rim of an automobile wheel, whose geometry description was given by T-spline surfaces. The examples demonstrated the potential of the immersed boundary approach for a full automation of the discretization process irrespective of the geometric complexity involved. Furthermore, we showed that hierarchical refinement can considerably increase the flexibility of immersed boundary methods in terms of adaptive resolution of local features in geometry and solution fields, while due to its simplicity and straightforward implementation, the key advantage of automated mesh generation for complex geometries can be fully maintained. We therefore believe that immersed boundary methods can open the door for a seamless isogeometric design-through-analysis procedure for complex engineering parts and assemblies.

Nevertheless, considerable work remains to be done. Some particular topics that need to

be investigated are the performance optimization of adaptive integration schemes, which could dramatically increase the computational efficiency, the analysis of topology changes and moving boundaries, for which immersed boundary methods offer potential advantages over ALE-type approaches, and the introduction of immersed boundary coupling schemes for multiphysics problems. It will be also necessary to perform careful studies of a variety of immersed boundary and interface problems to determine the resolution requirements of hierarchically refined NURBS to attain sufficiently accurate quantities of engineering interest. Further comparisons should also be made between surface-fitted unstructured meshes and immersed T-spline models to determine trade-offs. In addition, consideration should be given to engineering parts and systems of increased complexity that are very difficult to mesh in traditional ways. There will always be situations where surface-fitted three-dimensional meshing is preferable, but we believe that the present developments have provided a viable alternative pathway for many engineering design and analysis problems.

Acknowledgments. This work was accomplished during a three month visit of Dominik Schillinger at the Institute of Computational Engineering and Sciences (ICES) in summer 2011, for which financial support from the Centre of Advanced Computing (MAC) and the International Graduate School of Science and Engineering (IGSSE) of the Technische Universität München is gratefully acknowledged.

The ICES team gratefully acknowledges the support of the following research grants and contracts: ONR Grant N00014-08-1-0992, ARO contract W911NF-10-1-216, NSF GOALI CMI-0700807/0700204, NSF CMMI-1101007 and a grant from SINTEF. John Evans was partially supported by a DOE Computational Science Graduate Fellowship provided under grant DE-FG02-97ER25308, and Michael Scott was partially supported by an ICES CAM Graduate Fellowship.

The authors also acknowledge the Texas Advanced Computing Center (TACC) of the University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper (URL: <http://www.tacc.utexas.edu>). The authors also thank Martin Ruess for his help with the modal analysis of the propeller.

References

- [1] Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 2005; **194**:4135–4195.
- [2] Kagan P, Fischer A. Integrated mechanically based CAE system using B-spline finite elements. *Computer Aided Design*, 2000; **32**(8-9):539–552.
- [3] Cohen E, Martin T, Kirby RM, Lyche T, Riesenfeld R. Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:335–356.
- [4] Schmidt R, Kiendl J, Bletzinger KU, Wüchner R. Realization of an integrated structural design process: analysis-suitable geometric modelling and isogeometric analysis. *Computing and Visualization in Science*, 2010; **13**(7):315–330.

- [5] Cottrell JA, Hughes TJR, Bazilevs Y. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley: New York, 2009.
- [6] Evans JA, Bazilevs Y, Babuška I, Hughes TJR. n-Widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering*, 2009; **198**(21–26):1726–1741.
- [7] Cottrell JA, Reali A, Bazilevs Y, Hughes TJR. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 2006; **195**:5257–5296.
- [8] Elguedj T, Bazilevs Y, Calo VM, Hughes TJR. \bar{B} and \bar{F} projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Computer Methods in Applied Mechanics and Engineering*, 2008; **197**:2732–2762.
- [9] Auricchio F, da Veiga LB, Lovadina C, Reali A. The importance of the exact satisfaction of the incompressibility constraint in nonlinear elasticity: mixed fems versus NURBS-based approximations. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:314–323.
- [10] Taylor RL. Isogeometric analysis of nearly incompressible solids. *International Journal for Numerical Methods in Engineering*, 2011; **87**(1-5):273–288.
- [11] Kiendl J, Bletzinger KU, Linhard J, Wüchner R. Isogeometric shell analysis with Kirchhoff-Love elements. *Computer Methods in Applied Mechanics and Engineering*, 2009; **198**(49-52):3902–3914.
- [12] Benson DJ, Bazilevs Y, Hsu MC, Hughes TJR. Isogeometric shell analysis: The Reissner-Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:276–289.
- [13] Echter R, Bischoff M. Numerical efficiency, locking and unlocking of NURBS finite elements. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:374–382.
- [14] Bazilevs Y, Calo VM, Hughes TJR, Zhang Y. Isogeometric fluid-structure interaction: theory, algorithms and computations. *Computational Mechanics*, 2008; **43**:3–37.
- [15] Bazilevs Y, Hsu MC, Kiendl J, Wuechner R, Bletzinger KU. 3D Simulation of Wind Turbine Rotors at Full Scale. Part II: Fluid-Structure Interaction. *International Journal of Numerical Methods in Fluids*, 2011; **65**:236–253.
- [16] Bazilevs Y, Calo VM, Cottrell JA, Hughes TJR, Reali A, Scovazzi G. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 2007; **197**:173–201.
- [17] Akkerman I, Bazilevs Y, Calo VM, Hughes TJR, Hulshoff S. The role of continuity in residual-based variational multiscale modeling of turbulence. *Computational Mechanics*, 2008; **41**:371–378.
- [18] Bazilevs Y, Akkerman I. Large eddy simulation of turbulent Taylor-Couette flow using isogeometric analysis and residual-based variational multiscale method. *Journal of Computational Physics*, 2010; **229**:3402–3414.
- [19] Gomez H, Calo VM, Bazilevs Y, Hughes TJR. Isogeometric analysis of the Cahn-Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 2008; **197**:4333–4352.
- [20] Borden MJ, Verhoosel CV, Scott MA, Hughes TJR, Landis CM. A phase-field description of dynamic brittle fracture. Preprint submitted to *Computer Methods in Applied Mechanics and Engineering*, 2011.
- [21] Temizer I, Wriggers P, Hughes TJR. Contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering*, 2011; **200**:1100–1112.
- [22] Temizer I, Wriggers P, Hughes TJR. Three-dimensional mortar-based frictional contact treatment in isogeometric analysis with NURBS. *ICES REPORT 11-16*, The University of Texas at Austin, May 2011.
- [23] Verhoosel CV, Scott MA, Borden MJ, Hughes TJR, de Borst R. Discretization of higher-order gradient damage models using isogeometric finite elements. *ICES REPORT 11-12*, The University of Texas at Austin, May 2011.
- [24] Wall WA, Frenzel MA, Cyron C. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 2008; **197**:2976–2988.
- [25] Dedè L, Borden MJ, Hughes TJR. Isogeometric Analysis for topology optimization with a phase field model. *ICES REPORT 11-29*, The University of Texas at Austin, September 2011.

- [26] Borden MJ, Scott MA, Evans JA, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 2011; **87**:15–47.
- [27] Scott MA, Borden MJ, Verhoosel CV, Sederberg TW, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 2011; **88**:126–156.
- [28] Sederberg TW, Zheng J, Bakenov A, Nasri A. T-splines and T-NURCCSs. *ACM Transactions on Graphics*, 2003; **22**(3):477–484.
- [29] Sederberg TW, Cardon DL, Finnigan GT, North NS, Zheng J, Lyche T. T-spline simplification and local refinement. *ACM Transactions on Graphics*, 2004; **23**(3):276–283.
- [30] Scott MA, Li X, Sederberg TW, Hughes TJR. Local Refinement of Analysis-Suitable T-splines. *ICES REPORT 11-06*, The University of Texas at Austin, March 2011.
- [31] Forsey D, Bartels RH. Hierarchical B-Spline refinement. *Computer Graphics* (SIGGRAPH '88 Proceedings) 1988; **22**(4):205–212.
- [32] Forsey D, Bartels RH. Surface fitting with hierarchical B-Splines. *ACM Transactions on Graphics*, 1995; **14**(2):134–161.
- [33] Kraft R. Adaptive and linearly independent multilevel B-splines. In: *Surface Fitting and Multiresolution Methods*, Méhauté AL, Rabut C, Schumaker LL (eds). Vanderbilt University Press, 1997; 209–218.
- [34] Höllig K, Reif U, Wipperf J. Weighted extended B-spline approximation of Dirichlet problems. *SIAM Journal on Numerical Analysis*, 2001; **39**:442–462.
- [35] Höllig K. *Finite Element Methods with B-Splines*. Society for Industrial and Applied Mathematics: Philadelphia, 2003.
- [36] Schillinger D, Rank E. An unfitted *hp*-adaptive finite element method based on hierarchical B-splines for interface problems of complex geometry. *Computer Methods in Applied Mechanics and Engineering*, 2011; **200**(47-48):3358–3380.
- [37] Vuong AV, Giannelli C, Jüttler B, Simeon B. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 2011; **200**(49-52):3554–3567.
- [38] Zorin D, Schröder P, DeRose T, Kobbelt L, Levin A, Sweldens W. *Subdivision for Modeling and Animation*. SIGGRAPH course notes, 2000.
- [39] Warren J, Weimer H. *Subdivision Methods for Geometric Design*. Morgan Kaufman Publishers: San Francisco, 2002.
- [40] Samet H. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann Publishers: San Francisco, 2006.
- [41] Burstedde C, Wilcox LC, Ghattas O. p4est: Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees. *SIAM Journal on Scientific Computing*, 2011; **33**(3):1103–1133.
- [42] Bangerth W, Burstedde C, Heister T, Kronbichler M. Algorithms and Data Structures for Massively Parallel Generic Adaptive Finite Element Codes. *ACM Transactions on Mathematical Software*, 2011 (in press).
- [43] Yserantant H. On the multi-level splitting of finite element spaces. *Numerische Mathematik*, 1986; **49**:379–412.
- [44] Grinspun E, Krysl P, Schröder P. CHARMS: A Simple Framework for Adaptive Simulation. *ACM Transactions on Graphics*, 2002; **21**(3):281–290.
- [45] Krysl P, Grinspun E, Schröder P. Natural hierarchical refinement for finite element methods. *International Journal for Numerical Methods in Engineering*, 2003; **56**:1109–1124.
- [46] Bungartz HJ, Griebel M. Sparse grids. *Acta Numerica*, 2004; **13**:147–269.
- [47] Li X, Zheng J, Sederberg TW, Hughes TJR, Scott MA. On Linear Independence of T-splines. *ICES REPORT 10-40*, The University of Texas at Austin, October 2010.
- [48] Li X, Scott M. On the Nesting Behavior of T-splines. *ICES REPORT 11-13*, The University of Texas at Austin, May 2011.

- [49] Deng J, Chen F, Li X, Hu C, Tong W, Yang Z, Feng Y. Polynomial splines over hierarchical T-meshes. *Graphical Models*, 2008; **70**:76–86.
- [50] Li X, Deng J, Chen F. Polynomial splines over general T-meshes. *Visual Computer*, 2010; **26**:277–286.
- [51] Nguyen-Thanh N, Nguyen-Xuan H, Bordas SP, Rabczuk T. Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. *Computer Methods in Applied Mechanics and Engineering*, 2011; **200**(21-22):1892–1908.
- [52] Nguyen-Thanh N, Kiendl J, Nguyen-Xuan H, Wüchner R, Bletzinger KU, Bazilevs Y, Rabczuk T. Rotation free isogeometric thin shell analysis using PHT-splines. *Computer Methods in Applied Mechanics and Engineering*, 2011; **200**(47-48):4310–4324.
- [53] Sederberg TW, Finnigan GT, Li X, Lin H. Watertight trimmed NURBS. *ACM Transactions on Graphics*, 2008; **27**:1–79.
- [54] Wang W, Zhang Y, Scott MA, Hughes TJR. Converting an Unstructured Quadrilateral Mesh to a Standard T-spline Surface. *Computational Mechanics*, 2011; **48**(4):477–498.
- [55] Zhang Y, Wang W, Xu G, Hughes TJR. Converting an Unstructured Quadrilateral/Hexahedral Mesh to a Rational T-spline. *ICES REPORT 11-27*, The University of Texas at Austin, August 2011.
- [56] Zhang Y, Wang W, Hughes TJR. Solid T-spline Construction from Boundary Representations for Genus-Zero Geometry. *ICES REPORT 11-50*, The University of Texas at Austin, November 2011.
- [57] Li Z, Ito K. *The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains*. Society for Industrial and Applied Mathematics: Philadelphia, 2006.
- [58] Parvizian J, Düster A, Rank E. Finite Cell Method: h - and p - extension for embedded domain methods in solid mechanics. *Computational Mechanics*, 2007; **41**:122–133.
- [59] Düster A, Parvizian J, Yang Z, Rank E. The Finite Cell Method for Three-Dimensional Problems of Solid Mechanics. *Computer Methods in Applied Mechanics and Engineering*, 2008; **197**:3768–3782.
- [60] Schillinger D, Ruess M, Zander N, Bazilevs Y, Düster A, Rank E. Small and large deformation analysis with the p - and B-spline versions of the Finite Cell Method. Preprint submitted to *Computational Mechanics*, 2011.
- [61] Rank E, Kollmannsberger S, Sorger C, Düster A. Shell Finite Cell Method: A High Order Fictitious Domain Approach for Thin-Walled Structures. *Computer Methods in Applied Mechanics and Engineering*, 2011; **200**:3200–3209.
- [62] Rank E, Ruess M, Kollmannsberger S, Schillinger D, Düster A. Geometric modeling, Isogeometric Analysis and the Finite Cell Method. Preprint submitted to *Computer Methods in Applied Mechanics and Engineering*, 2011.
- [63] Piegl LA, Tiller W. *The NURBS Book (Monographs in Visual Communication)*. Springer, New York, 1997.
- [64] Rogers DF. *An Introduction to NURBS with Historical Perspective*. Morgan Kaufmann Publishers: San Francisco, 2001.
- [65] Farin G. *Curves and surfaces for CAGD*. Morgan Kaufmann Publishers: San Francisco, 2002.
- [66] Rhino. CAD modeling and design toolkit. www.rhino3d.com (24th Sept 2011)
- [67] Chui CK. *An introduction to wavelets*. Academic Press: London, 1992.
- [68] Peters J, Reif U. *Subdivision surfaces*. Springer: Heidelberg, 2008.
- [69] Sabin M. *Analysis and Design of Univariate Subdivision Schemes*. Springer: Heidelberg, 2010.
- [70] Rank E. Adaptive remeshing and h - p domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 1992; **101**:299–313.
- [71] Schillinger D, Düster A, Rank E. The hp - d adaptive Finite Cell Method for Geometrically Nonlinear Problems of Solid Mechanics. *International Journal for Numerical Methods in Engineering*, 2011; doi:10.1002/nme.3289.
- [72] Ainsworth M, Oden JT. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 1997. **142**:1–88.
- [73] Zienkiewicz OC, Taylor RL. *The Finite Element Method Vol. 3: Fluid Dynamics*. Butterworth-Heinemann: Oxford, 2000.

- [74] Donea J, Huerta A. *Finite Element Methods for Flow Problems*. Wiley: Chichester, 2003.
- [75] Brooks AN, Hughes TJR. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 1982; **32**:199–259.
- [76] Zhu T, Atluri SN. A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. *Computational Mechanics*, 1998; **21**:211–222.
- [77] Fernandez-Mendez S, Huerta A. Imposing essential boundary conditions in mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 2004; **193**:1257–1275.
- [78] Golub G, Van Loan CF. *Matrix Computations*. Johns Hopkins University Press: Baltimore, 1996.
- [79] Bazilevs Y, Hughes TJR. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 2007; **36**:12–26.
- [80] Embar A, Dolbow J, Harari I. Imposing Dirichlet boundary conditions with Nitsche’s method and spline-based finite elements. *International Journal for Numerical Methods in Engineering*, 2010; **83**:877–898.
- [81] Hansbo A, Hansbo P. An unfitted finite element method, based on Nitsche’s method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering*, 2002; **191**:537–552.
- [82] Evans JA, Hughes TJR. Explicit trace inequalities for isogeometric analysis and parametric finite elements. *ICES REPORT 11-17*, The University of Texas at Austin, May 2011.
- [83] Govindjee S, Strain J, Mitchell TJ, Taylor R. Convergence of an Efficient Local Least-Squares Fitting Method for Bases with Compact Support. Manuscript submitted to *Computer Methods in Applied Mechanics and Engineering*, 2011.
- [84] Mundani RP, Bungartz HJ, Rank E, Niggel A, Romberg R. Extending the p -Version of Finite Elements by an Octree-Based Hierarchy. In: Widlund OB, Keyes DE (eds.), *Lecture Notes in Computational Science and Engineering*, Vol. 55, Springer: New York, 2006, pp. 699–706.
- [85] Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 2005; **37**:239–261.
- [86] Chen W, Cai Y, Zheng J. Generalized hierarchical NURBS for interactive shape modification. In: *Proceedings of the 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, 2008.
- [87] Chen W, Cai Y, Zheng J. Freeform-based form feature modeling using a hierarchical & multi-resolution NURBS method. In: *Proceedings of the 9th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, 2010.
- [88] Trilinos Version 10.6. Sandia National Laboratories, <http://trilinos.sandia.gov/>, Los Alamos, NM, 2011.
- [89] Bazilevs Y, Calo VM, Cottrell JA, Evans JA, Hughes TJR, Lipton S, Scott MA, Sederberg TW. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:229–263.
- [90] Dörfel MR, Simeon B, Jüttler B. Adaptive isogeometric analysis by local h -refinement with T-splines. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:264–275.
- [91] Carey GF, Utku M. Stream function finite element solution of Stokes flow with penalties. *International Journal for Numerical Methods in Fluids*, 1984; **4**:1013–1025.
- [92] Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 1982; **48**:387–411.
- [93] Sueli E, Mayers DF. *An Introduction to Numerical Analysis*. Cambridge University Press: Cambridge, 2003.
- [94] MacNeal RH, Harder RL. A Proposed Standard Set of Problems To Test Finite. Element Accuracy. *Finite Elements in Analysis and Design*, 1985; **1**:3–20.
- [95] Belytschko T, Stolarski H, Liu WK, Carpenter N, Ong JSJ. Stress projection for membrane and shear locking in shell finite elements. *Computer Methods in Applied Mechanics and Engineering*, 1985; **51**:221–258.
- [96] Rank E, Düster A, Nübel V, Preusch K, Bruhns OT. High order finite elements for shells. *Computer Methods in Applied Mechanics and Engineering*, 2005; **194**:2494–2512.

- [97] Bischoff M, Wall WA, Bletzinger K-U, Ramm E. Models and finite elements for thin-walled structures. In: Stein E, de Borst R, Hughes TJR (eds). *Encyclopedia of Computational Mechanics, Vol. 2. Solids, Structures and Coupled Problems*, Wiley: New York, 2004.
- [98] Löhner R, Cebal R, Camelli FE, Appanaboyina S, Baum JD, Mestreau EL, Soto OA. Adaptive embedded and immersed unstructured grid techniques. *Computer Methods in Applied Mechanics and Engineering*, 2008; **197**:2173–2197.
- [99] Peskin C. The Immersed Boundary Method. *Acta Numerica*, 2002; **11**:1–39.
- [100] Glowinski R, Kuznetsov Y. Distributed Lagrange multipliers based on fictitious domain method for second order elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 2007; **196**:1498–1506.
- [101] Ramière I, Angot P, Belliard M. A fictitious domain approach with spread interface for elliptic problems with general boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 2007; **196**:766–781.
- [102] Dolbow J, Harari I. An efficient finite element method for embedded interface problems. *International Journal for Numerical Methods in Engineering*, 2009; **78**:229–252.
- [103] Gerstenberger A, Wall WA. An embedded Dirichlet formulation for 3D continua. *International Journal for Numerical Methods in Engineering*, 2010; **82**:537–563.
- [104] Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering*, 2010; **199**:2680–2686.
- [105] Burman E, Hansbo P. Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Applied Numerical Mathematics*, 2011; doi:10.1016/j.apnum.2011.01.008.
- [106] Sanches R, Bornemann P, Cirak F. Immersed B-spline (i-spline) finite element method for geometrically complex domains. *Computer Methods in Applied Mechanics and Engineering*, 2011; **200**: 1432–1445.
- [107] Rüberg T, Cirak F. Subdivision-stabilised immersed B-spline finite elements for moving boundary flows. Preprint submitted to *Computer Methods in Applied Mechanics and Engineering*, 2011.
- [108] Schillinger D, Kollmannsberger S, Mundani R-P, Rank E. The finite cell method for geometrically nonlinear problems of solid mechanics. *IOP Conference Series: Material Science and Engineering*, 2010. **10**:012170.
- [109] Sadd MH. *Elasticity. Theory, Applications, and Numerics*. Academic Press: Oxford, 2009.
- [110] Hughes TJR. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover: New York, 2000.
- [111] Zienkiewicz OC, Taylor RL. *The Finite Element Method Vol. 1: The Basis*. Butterworth-Heinemann: Oxford, 2000.
- [112] T-Splines, Inc. Surface modeling software based on T-splines. www.tsplines.com (24th Sept 2011)
- [113] Bungartz HJ, Griebel M, Zenger C. *Introduction to Computer Graphics*. Charles River Media: Hingham, 2004.
- [114] Common Versatile Multi-purpose Library for C++, developed at the University of Geneva; <http://tech.unige.ch/cvmlcpp/> (October 13th, 2011)
- [115] de Berg M, Cheong O, van Kreveld M, Overmars M. *Computational Geometry: Algorithms and Applications*. Springer: Heidelberg, 2008.
- [116] Bindick S, Stiebler M, Krafczyk M. Fast kd-tree-based hierarchical radiosity for radiative heat transport problem. *International Journal for Numerical Methods in Engineering*, 2011; **86**:1082–1100.
- [117] ARPACK. Software for the solution of large scale eigenvalue problems developed by R. Lehouq, D. Sorensen, K. Maschhoff, C. Yang; <http://www.caam.rice.edu/software/ARPACK/> (October 11th, 2011)
- [118] MKL PARDISO. Direct solver contained in Intel’s MKL library; www.intel.com/software/products/mkl (16th January 2012)